PRISM Driver Programmers Manual

Development Information

For distribution under a Non-Disclosure Agreement only.

This document instructs the host programmer or device driver writer in how to use the services provided by the HFA3841, HFA3842, or future PRISM MACs, and provides a reference to all commands and modes of operation.

The intended audience for this manual is programmers, device driver writers, application engineers, and WLAN adapter hardware designers.

Revision History

Version	Item	Date	Contents
0.11	(none)	1998-03-16	Advance information
1.00	(none)	1998-0407	Initial release
1.10	(none)	1998-06-26	Updates and new material
1.20	(none)	1998-07-16	Updates and new material
1.30a	CT000559	1998-07-31	References
1.50	CT000573	1998-02-24	New content: firmware description
1.61	AN9900.0	2000-06-09	Reorganized and updated with new API documentation
1.70	AN9900.1	2000-11-29	Incorporate new RIDs and updated enhancements; add chapters for USB and PCI implementations
2.00	AN9900.2	2001-05-16	Reformatted and revised for grammar, style, and clarity. Major revisions of Chapters 3, 4, 6, and 7 and Appendix F.
2.01	AN9900.2	2001-06-20	Corrected page numbering; expanded information on CommsQuality and encryption/decryption in Chapter 4.
2.10	AN9900.2	2001-08-31	Added description of Monitor Mode (Appendix F) and Promiscuous Mode (Chapter 4); added description of APIs to support Windows XP and NDIS 5.1 operation
2.20	AN9900.2	2002-01-31	Added Appendix A as complete, cross-referenced list of RIDs; incremented numbering of remaining appedixes. Extensively revised Appendix B, NDIS Driver API.
2.30	RM025	2002-6-24	Reformatted to new corporate documentation standard. Added description of AP security enhancements (RIDs FC29, FCB7, FCB8, FCB9, F204) and fallback control (RID FCB5). Corrected description of RID FCB6. Corrected organization of static vs dynamic RIDS. Added definition of Error Qualifiers and Passive Scan Control (Chapter 4) and PDRs 405, 407, 410, and 411 (Chapter 7). Deleted Appendix E (Pending API Enhancements) and renumbered remaining appendixes.

Notes:

All Intersil U.S. products are manufactured, assembled and tested utilizing ISO9000 quality systems.
Intersil Corporation's quality certifications can be viewed at www.intersil.com/design/quality

For information regarding Intersil Corporation and its products, see www.intersil.com

Sales Office Headquarters

NORTH AMERICA
Intersil Corporation
7585 Irvine Center Drive
Suite 100
Irvine, CA 92618
TEL: (949) 341-7000
FAX: (949) 341-7123

Intersil Corporation
2401 Palm Bay Rd.
Palm Bay, FL 32905
TEL: (321) 724-7000
FAX: (321) 724-7946

EUROPE
Intersil Europe Sarl
Ave. William Graisse, 3
1006 Lausanne
Switzerland
TEL: +41 21 6140560
FAX: +41 21 6140579

ASIA
Intersil Corporation
Unit 1804 18/F Guangdong Water Building
83 Austin Road
TST, Kowloon Hong Kong
TEL: +852 2723 6339
FAX: +852 2730 1433

Table of Contents

Appendixes

Tables

Figures

Chapter 1 About This Manual

1.1 Introduction

The PRISM™ Medium Access Controller (PRISM MAC) provides a set of services to the host computer or host driver software including:

- sending submitted data packets formatted either as 802.3 or 802.11 frames
- receiving and passing data packets formatted either as 802.3 or 802.11 frames
- reporting status changes such as association, authentication, and disassociation
- configuring various parameters of the operation of an 802.11 Wireless Local Area Network (WLAN) adapter
- installing new or updated functionality

1.2 Purpose

This document instructs the host programmer or device driver writer in how to use the services provided by the HFA3841, HFA3842, or future PRISM MACs, and provides a reference to all commands and modes of operation. The references and instructions propagate to all devices, except where it is expressly stated that they relate only to a specific device.

The intended audience for this manual is programmers, device driver writers, application engineers, and WLAN adapter hardware designers.

1.3 Reference Documents

Familiarity of the reader with the 802.3 and 802.11 standards is assumed. In particular, this document makes extensive use of abbreviations and acronyms used in the 802.11 standard document. For explanation of unfamiliar abbreviations or acronyms, please refer to the relevant standards documents.

[1] IEEE Std 802.11-1997, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, as revised by: Revisions to IEEE Std 802.11-1997, IEEE Draft 802.11rev/D10 (January 1999).

[2] IEEE Std 802.1H-1995, Recommended Practice for Medium Access Control (MAC) Bridging of Ethernet V2.0 in IEEE 802 Local Area Networks (with errata sheet).

[3] ISO/IEC 8802-3: 1993 (E), ANSI/IEEE Std 802.3, 1993 Edition.

[4] PC Card Standard Electrical Specification, PCMCIA/JEIDA, May 1996.

[5] HFA3841 Data Sheet, File Number 4661.2, Intersil Corporation, Melbourne, FL, Copyright 2000; Document #4661.2.

[6] NDIS MiniPort OID Interface, Microsoft, Windows 95/98/NT

[7] IEEE Std 802.11b/D7.0-1999, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Higher Speed PHY extension in the 2.4GHz band (July 1999)

[8] Universal Serial Bus Specification, Revision 1.1, September 9, 1998.

[9] PCI Local Bus Specification, Revision 2.1, June 1, 1995.

[10] HFA3842 Data Sheet, File Number 4839, Intersil Corporation, Melbourne, FL, Copyright 2000, Document #FN4839.

1.4 Definitions

The following sections provide definitions for terms and concepts unique to the implementation and operation of the PRISM MAC.

1.4.1 COMMANDS AND EVENTS

The host manages operation of the PRISM MAC through a set of *commands* issued through a *command register*. The PRISM MAC indicates results through *events* posted in an *event status register*. The PRISM MAC interface interlocks through a set of *busy bits* associated with the command register and through certain other command and parameter registers documented in this manual.

1.4.2 BUFFERS

The PRISM MAC is designed to move and process data in blocks called *buffers* in special-purpose memory. This memory is part of the *network interface card* (*NIC*) containing the PRISM MAC and the associated 802.11 Physical Layer (*PHY*).

1.4.3 BUFFER ACCESS PATH AND FRAME IDENTIFIERS

The host accesses buffers indirectly through a *Buffer Access Path* (*BAP*), which is a set of host-accessible registers that allow addressing, reading, and writing of areas of buffer memory designated by *Frame IDentifiers* (*FIDs*).

1.4.4 RESOURCE IDENTIFIERS

Host access to management and configuration data items is implemented by overloading the BAP mechanism using special values, called *Resource IDentifiers* (*RIDs*), for reading from and writing to a special-purpose buffer. Writing a RID to the BAP in the appropriate manner causes configuration information within the PRISM MAC memory to be copied to the special buffer, where it may be accessed by the host for reading or writing.

1.4.5 CONTROLLER AND FIRMWARE

The PRISM MAC is implemented using a proprietary *controller* designed for data movement and protocol execution, along with special-purpose hardware for event processing and memory management. Each of the 802.11 MAC protocol state machines, as well as the state machines for controlling the physical layer and responding to host commands, is *firmware* operating in one of eight prioritized independent execution contexts in the controller.

1.4.6 FRAME DESCRIPTOR

The first block of 128 bytes of an allocated buffer is used as a special data structure called a *Frame Descriptor* (*FD*). This structure is used internally by the firmware to maintain state information about a frame as it is in the process of being handled. Only part of the structure is visible to the host; this part is called a *Frame Structure* (*FS*). This host-visible part of the FD is used to communicate options and status information associated with individual frames between the driver and the firmware. Details of the invisible portion of the FD are not relevant to the driver programmer and are not included in this document.

1.4.7 MAC PORTS

The *MAC Port* is a logical port for incoming and outgoing frames from and to a specific logical link. The MAC Port is implemented as a separate field in the FD of frames transmitted and received over the host interface. By reading the MAC Port number of received frames and by writing the MAC Port number of frames that are to be transmitted, the driver can distinguish between frames handled by different MAC Ports.

1.4.8 PSEUDO-IBSS

As an implementation compromise and to facilitate early testing before the availability of 802.11 Access Points, a non-standard operating mode called *Pseudo-IBSS* was implemented in the PRISM MAC firmware. This operating mode allows two stations to communicate with each other without the presence of an Access Point, and without the distributed beaconing called for by the 802.11 Standard.

Warning! Pseudo-IBSS operation is unsupported. Use it only in non-customer configurations, and at your own risk.

1.4.9 MONITOR MODE

Monitor mode is a non-standard feature of the PRISM MAC firmware, intended for use only in testing and debugging environments. When a station is operating in Monitor mode, it receives and passes to the host all frames wherein the PLCP header and beginning of the frame are successfully received, regardless of FCS errors or other protocol defects.

Warning! Monitor mode is incompatible with operation as a Standard-compliant STA. In particular, it poses significant security risks in a production network. Intersil may remove Monitor mode from future releases of the firmware without advance notice. It does not participate in the 802.11 network, it only listens. Use it only at your own risk.

1.4.10 PRODUCTION DATA AREA

The *Production Data Area* (*PDA*) is an area of non-volatile storage written at manufacturing and calibration time with information about the configuration of a particular NIC. The PDA contains a number of *Production Data Records* (*PDRs*).

The primary use of a PDR is to provide reference data that is unique to the NIC. This information is used by downloading utilities to “plug” a firmware load image with values that are unique to the NIC prior to downloading operational firmware to the NIC.

The PDA is not used exclusively for plugging records. The PDA can be used to store information that is not used by the firmware image. For example, transitory manufacturing information could be inserted in the PDA and used by the various steps in the manufacturing process.

1.4.11 CONFIGURATION ENTITIES

The behavior of the PRISM MAC firmware depends on a hierarchy of *configuration entities* that are read at various stages in the operation of a NIC.

Fixed configuration entities are parameters whose values are provided by PDRs defined during production of the NIC (hardware and firmware). These values cannot be changed during operation. The following configuration information is set at manufacturing time:

- MAC Address
- Domain information
- Calibration information
- NIC Identity

Static configuration entities contain values that are copied by the synchronous Enable command process. These entities are set at initialization of the NIC and persist across POR cycles. The copied values influence the behavior of processes that are active only in Station Enabled state. Therefore, changing Static Configuration Entity values only has effect if followed by execution of the Enable command.

Dynamic configuration entities are set during operation and influence the behavior of the NIC. These entities contain values that directly influence the execution of processes active in Station Enabled state; changes in these entities take effect immediately.

Chapter 2 Driving a Station Card

Once primary functions are present in a card, station firmware is enabled by downloading the secondary (Station module) firmware component. Secondary firmware resides in non-volatile storage.

Driving a station card consists of the following major tasks, in addition to whatever adaptation to the host operating environment is necessary:

- initializing the card and firmware
- enabling reception and handling reception events
- performing buffer allocation and transmit sequences
- handling transmission events and, if required, transmit exceptions
- using the RID interface, Configuration Records, and Information Frames to manage the operation of the card as desired

2.1 Initialization

This section describes the necessary initialization for both the stand-alone PRISM MAC and the board containing the PRISM MAC.

After power-up or a reset, initiated by the host writing a value of 1 to bit 7 (SRESET) of the Configuration Option Register (COR), the PRISM MAC and PHY are initialized by the firmware in a predefined sequence. Table 2.1 presents this sequence

Table 2.1: Firmware and Hardware Initialization

Step	Host…	PRISM MAC…
1		clears PC Card Ready line
2		sets Command Busy bit
3		performs basic volatile memory test
4		checks memory configuration
5		initializes I/O register memory area
6		copies CIS from non-volatile storage into Attribute RAM area
7		asserts PC Card Ready line to show initialization complete
8	reads the CIS	
9		builds necessary jump/thread tables in RAM
10		initializes RAM data
11		initializes context-specific variables in RAM
12		copies configuration parameters from non-volatile storage into RAM
13		copies additional firmware from non-volatile NV storage into RAM
14		initializes additional firmware
15		waits for host to switch to I/O mode (wait forever)
16	sets the PRISM MAC into I/O mode by writing a value of 1 to bit 0 of the COR	

Table 2.1: Firmware and Hardware Initialization (Continued)

17		copies PHY configuration parameters from non-volatile storage to the PHY
18		configures other PHY circuits
19		initializes PRISM MAC contexts (contexts 1 and 2)
20		builds internal buffer structure
21		clears CmdBusy bit
22		clears OffsetBusy bits

2.1.1 INITIALIZATION PROCESS DESCRIPTION

The steps in the initialization sequence are described in the following paragraphs.

Clear PC Card Ready line: This is done by the PRISM MAC within 25 µsec after power-up or reset, in order to signal the host that the initialization might take more than 20 msec. (Reference [4], Vol 2, 4.4.6). The hardware must clear this line upon power-up or reset.

Set Command Busy bit: The hardware sets the Command Busy bit.

Perform basic memory test: The basic memory test simply verifies that the connections to the RAM are not shorted or open. If the memory check fails, the firmware stops and the PRISM MAC does not react to any external stimulus, except for power down/up and hardware reset.

Check memory configuration: Check what type of non-volatile storage is used.

Initialize I/O register memory area: To all zeroes.

Copy CIS from non-volatile storage into Attribute RAM area: After this step, these values are available to be read by the host.

Assert PC Card Ready line to show CIS initialization is complete: After detecting the asserted Ready line, the host decides, based on the CIS values, whether it can support the power requirements of the NIC. If not, the host does not switch the NIC into the I/O card mode.

NOTE: The CIS must be made available before initializing the PHY, since the PHY power requirements typically far exceed the maximum values given in the PC Card standard.

Build necessary jump/thread tables in RAM: Since part of the firmware in the PRISM MAC resides in masked or FLASH ROM and some of this firmware must be replaceable, these parts are invoked through a jump table in RAM. To make use of that mechanism, these tables must be initialized.

Initialize RAM data: Necessary RAM data items are initialized with their required values.

Initialize context-specific variables in RAM: All context-specific variables are either zeroed or copied from non-volatile memory.

Copy configuration parameters from NV storage into RAM: All the changeable configuration parameters are kept in a RAM image of the non-volatile storage. This image must be copied from non-volatile storage.

Copy additional firmware from NV storage into RAM: Part of the firmware, such as additions and bug fixes, can reside in non-volatile storage. This firmware can be executed from non-volatile storage, but with a speed penalty. To have parts of the firmware execute at higher speed, they must be copied into RAM.

Initialize additional firmware: The copied firmware must be initialized and the related jump vectors must be updated to allow integration of the additional firmware.

Wait for host to switch to I/O card: The PRISM MAC waits to be configured as an I/O card by the host. When the host configures the NIC as a memory card, the PRISM MAC stays in this wait mode.

Copy PHY configuration parameters from NV storage to PHY: At power-up or reset, the PHY requires a set of parameters to be downloaded. The PRISM MAC copies the information from non-volatile storage to the PHY, using the pre-defined Modem Management Interface (MMI), and places the PHY in sleep mode.

Configure other PHY circuits: Other integrated circuits, such as frequency synthesizers, must be initialized by the PRISM MAC. The data from non-volatile storage is copied transparently to the related integrated circuits.

Initialize context hardware: The hardware of every context is initialized to its correct state.

Build internal buffer structure: The internal buffer structure consists of a number of 128-byte blocks that are linked together. This buffer structure must be initialized.

Clear CmdBusy and OffsetBusy(s): Clear the busy signals to place these bits in a defined state.

2.1.2 COMMAND STATE DIAGRAM

Once the initialization sequence completes, the PRISM MAC is ready to accept commands and data. After reset, only a limited number of commands are accepted; other commands are only accepted after enabling the communications. Figure 2.1 illustrates where and when the commands (shown in square brackets) can occur.

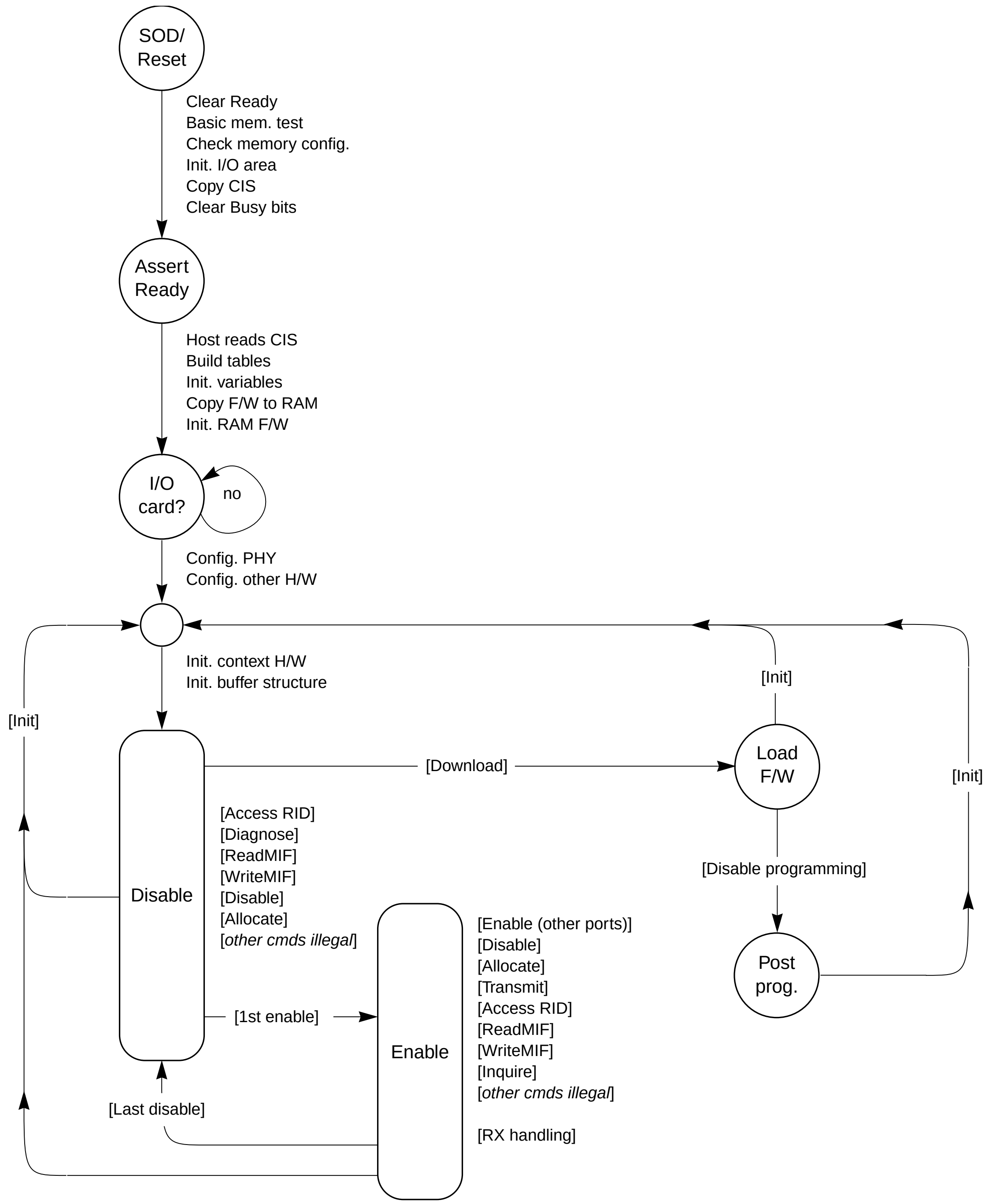


Figure 2.1: PRISM MAC Command State Diagram

2.2 Basic Operations

The following sections describe basic operations of the MAC in the command state.

2.2.1 PROGRAMMING AFTER TRANSITION TO I/O MODE

After the Card Services module or PCMCIA enabler has issued the command to place the card in I/O mode, initialization as a station consists of the following interactions:

Operation	Host…	PRISM MAC…
Wait for readiness	Waits until the Busy bit in the Command Register is clear.	Clears the Busy bit.
Initialization	Issues the command to initialize the PRISM MAC.	Firmware carves up available NIC memory into a linked list of 128-byte blocks identified by pointers known as Frame Identifiers (FIDs) and enters the Disabled state.
	Waits until the Busy bit in the Command Register is clear.	Clears the Busy bit.
Configure the PRISM MAC for STA operation	Sets the MAC port type to 1 by writing to RID 0xFC00.	Receives the command and updates the configuration item.
	Waits for the Busy bit to clear before proceeding.	Clears the Busy bit.
	Sets the desired SSID by writing to RID 0xFC02.	Receives the command and updates the configuration item.
	Waits for the Busy bit to clear before proceeding.	Clears the Busy bit.
	Sets the desired maximum data length by writing to RID 0xFC07.	Receives the command and updates the configuration item.
	Waits for the Busy bit to clear before proceeding.	Clears the Busy bit.
	Sets other desired configuration items by writing to the appropriate RIDs.	Receives the command and updates the configuration item.
	Waits for the Busy bit to clear before proceeding.	Clears the Busy bit.
Establish event handler(s)	Sets the desired bit(s) in the Interrupt Enable register to map processing of the Events to the appropriate interrupt handling function(s). **NOTE**: The host must acknowledge events that are holding resources, such as Rx, Tx, TXEL, Info, Alloc, Cmd.	
Enable operation	Issues the Enable command	Firmware begins searching for the configured SSID by issuing probe requests on the configured channel(s), attempts to authenticate and associate with the desired BSS if found, and passes status change and received date frame events to the host.

Once the NIC is in the Enabled state, normal receive and transmit operations involve buffer manipulation.

2.2.2 BUFFER MANIPULATION

Buffer manipulation in the PRISM MAC is efficient, with both hardware and firmware support for manipulating the buffers.

Operation	Host…	PRISM MAC…
Buffer allocation	Requests allocation of a buffer.	Firmware walks the free list until it gets enough blocks to satisfy the request. **NOTE**: The allocation process does *not* look for a free contiguous area of memory to satisfy the request; all buffer blocks are the same 128-byte size. Instead, the firmware simply retrieves the number of linked blocks needed to satisfy the request. This allows the allocator to quickly get the requested memory (if there are enough free blocks).
		Firmware returns a pointer to the first block to the host via an allocation event, referred to as the FID.
Allocation event processing	Saves the FID indicated in the allocation event to use as a token to refer to the allocated buffer, then acknowledges the allocation event.	
Buffer access	Places the FID in the BAP select register.	
	Writes the logical offset into the buffer to the BAP offset register.	Hardware sets Busy bit. Firmware receives a host attention event, calculates the buffer pointer and physical offset, and prepares to access the buffer. Sets up hardware support registers.
	Waits for the Busy bit to be cleared before proceeding.	Firmware acknowledges host attention event, causing the Busy bit of the BAP offset register to be Cleared.
Buffer read	Reads the BAP data register.	Hardware reads the indicated location into the BAP data register, and increments the physical pointer to the next logical buffer location.
Buffer write	Writes a value to the BAP data register.	Hardware reads the BAP data register, writes to the current physical location, and increments the physical pointer to the next logical buffer location.

2.2.3 PRISM MAC RECEIVE

State or Event	Host…	PRISM MAC…
Ready to receive		Firmware maintains a pool of pre-allocated maximum size buffers for handling the received packets. Whenever the pre-allocated receive queue falls below a threshold, a background thread in the firmware starts to allocate additional receive buffers.
Packet received		Places the received data into the first receive buffer on the queue. IF the packet is valid, Generate the ACK if required; Truncate the buffer, returning unused 128-byte blocks to the free pool; Report the receive to the host with a RECEIVE event in EvStat. **NOTE**: A write followed by a read will not access the same memory location, but instead will access sequential memory locations. Every read or write access of the Data register will cause the hardware to increment the physical pointer and routes the packet according to type: Data Frames to the host, Management and Controller to handlers in firmware.
RECEIVE event	Retrieves the FID from the RxFID register Places the FID in the BAP select register.	Writes the logical offset into the buffer to the BAP offset register. Firmware receives a host attention event, calculates the buffer pointer and physical offset, and prepares to access the buffer.
Buffer read	Repeatedly reads the BAP data register until all the data are retrieved into host buffers. **NOTE**: The number of times to read can be determined from the data length in the frame structure, which is in the first buffer presented.	Hardware reads the indicated location into the BAP data register, and increments the physical pointer to the next logical buffer location on each read.
Receive event acknowledgement	Writes to the EvAck register indicating acknowledgment of the receive event. **NOTE**: As long as the host is acknowledging these receive events, the PRISM MAC receiver should never be backed up due to lack of receive buffers.	Receives a host attention event; returns the buffers in the list to the system free buffer pool.

2.2.4 PRISM MAC TRANSMIT

State or Event	Host…	PRISM MAC…
Buffer allocation	Before each transmit, requests an allocation of a buffer.	Firmware receives a host attention event; walks the free list until it gets enough blocks to satisfy the request.
	Saves the FID indicated in the allocation event to use as a token to refer to the allocated buffer.	Firmware returns a pointer to the first block to the host via an allocation event.
OR Buffer reservation	Uses the more efficient method of reserving a pool of FIDs and performing the transmits with the Reclaim bit set. Reclaim causes the firmware to return the FID, with a new chain of buffers attached, to the host via an allocation event so that the host doesn't have to issue a separate Allocation command for each packet it wants to transmit.	Firmware responds to each allocation request with a different FID.
Buffer access	Places the FID in the BAP select register.	
	Writes the logical offset into the buffer to the BAP offset register.	Hardware sets Busy bit. Firmware receives a host attention event, calculates the buffer pointer and physical offset, and prepares to access the buffer. Sets up hardware support registers.
	Waits for the Busy bit to be Cleared before proceeding.	Firmware acknowledges host attention event, causing the Busy bit of the BAP offset register to be Cleared.
Buffer write	Writes packet data to the BAP data register.	For each write, hardware reads the BAP data register; writes to the current physical location; and increments the physical pointer to the next logical buffer location.
Transmit command	Writes transmit FID to Param0 register. Writes Transmit command to Command register.	Firmware receives a host attention event and puts the frame referenced by the FID on the transmit queue for further processing. Firmware waits for transmit to complete; services other events that occur in the meantime.
	Waits for interrupt or polls EVSTAT.	Firmware acknowledges frame transmission with TX bit in EVSTAT, OR indicates failure with TXEXC in EVSTAT.
Transmit complete	Acknowledges receipt of status by writing the corresponding bit in the EvAck register.	
Alloc event	Saves the FID indicated in the allocation event to use as a token to refer to the reclaimed buffer.	*IF* the Transmit command indicated Reclaim, and the buffer is no longer needed by the firmware, *THEN* firmware ensures the FID has the appropriate number of buffers linked (pulled from the free list), and returns a pointer to the first block to the host via an allocation event. *OTHERWISE*, if the Transmit command did not indicate Reclaim, the host must get another buffer through an Allocation command.

2.2.5 NUMBER OF FIDS TO RESERVE

Intersil does not specify what the correct number of pre-allocated FIDs should be, nor is there data available for this pre-allocation at the release of this document. There is no necessary correlation between the number of outstanding transmits and the number of pre-allocated FIDs when the Reclaim bit is used. Beyond a small number, preallocating FIDs has no value because the Reclaim mechanism will usually return the same FID, with a new buffer chain attached, before the original buffer has been completely transmitted.

2.2.6 ORDER OF EVENTS

The reclaim will complete when the background thread has successfully allocated the required number of buffers. Typically, the reclaim completes (and is reported to the host as an allocation event) before the PRISM MAC has finished transmitting the packet. Because of this, the host can safely have more posted transmits than pre-allocated FIDs. The reference NDIS miniport driver pre-allocates three FIDs. The use of the Reclaim mechanism implies that the host may not depend on the FID as a key to specific buffer contents associated with a FID that is returned by a TxComplete or TxExc event. Instead, should the host desire to identify successful or unsuccessful transmitted frames and associate them with their contents, the host must use the SwSupport field (see 4.3.2.4.2 SwSupport Field (word offset 3)) in the Communication Frame Structure.

The SWSupport field is available in the Transmit (Tx) Frame Structure to stamp an identifier on the frame. These fields are not used by the firmware and their value will be maintained when returned back to the host.

2.2.7 BUFFER ALLOCATION PRIORITY

The priority for buffer allocation requests in the PRISM MAC is to maintain

- first, receive queue buffers;
- second, transmit with reclaim request, and
- last, Allocation command request.

This priority helps ensure the receive handler is always able to handle incoming packets (assuming the host is processing the receive events).

Chapter 3 Driving an Access Point

Driving a card in an access point consists of the same basic operations as driving a card in a station, with some additions. These additions are dependent on which of the two major operating modes or access point support packages is implemented: firmware-based or host-based. These modes are detailed in the following sections.

Note also that for roaming between Access Points to work, the developer must implement an Inter-Access Point Protocol. This is a requirement for both firmware-based and host-based Access Points. This requirement is discussed in the final section of this chapter.

3.1 Firmware-based AP Mode

Firmware-based AP mode is enabled by downloading an additional firmware component, called a *tertiary* module, into RAM. These functions may, in principle, be resident in non-volatile storage or downloaded into RAM at reset time. Tertiary functions are downloaded into RAM at system startup time.

The tertiary module handles most of the real-time response and control functions of an 802.11 access point. These include authentication, association, and forwarding of directed frames received at the access point (and which are addressed to other stations in the BSS). Depending on configuration, the firmware passes to the host only those received frames either directed to the host’s MAC address or destined for addresses not in the BSS (and which are to be bridged to the distribution system).

Driving a firmware-based AP card in an access point consists of the same basic operations as driving a card in a station plus some additions. Special configuration items are set, and the access point driver and software accomplish the following additional major tasks, in addition to whatever adaptation to the host operating environment is necessary:

- downloading the tertiary functions and initializing the card. Note that if the card is reset, the contents of RAM are not preserved, and this sequence will have to be repeated.
- setting the BSS ID string
- setting the default channel upon which the access point is to allow associations
- setting supported and basic transmission rates
- enabling the card
- handling received frames and performing the appropriate address filtering and bridging, forwarding, or routing to the distribution system
- handling frames forwarded from the distribution system and transmitting them to the appropriate station
- using the RID interface, Configuration Records, and Information Frames to provide the required management interface to network management software

3.1.1 BASIC INITIALIZATION

Table 3.1 describes basic initialization operations for firmware-based AP mode.

After the Enable command is issued, operation is similar to a station with respect to reception, transmission, and buffer management. The host of a firmware-based AP will receive information frames not seen by a station driver and should record and act on the information appropriately.

3.2 Host-based AP Mode

In host-based AP mode, the functions that perform many of the operations maintaining a BSS are not present in the firmware. Instead, the host software performs such functions as:

- association and authentication
- reception, filtering, and retransmission of frames that are addressed to other members of the BSS
- all power-save queuing
- other management and data transmission operations as appropriate

Table 3.1: Basic Initialization in Firmware-based AP Mode

Operation	Host…	PRISM MAC…
Wait for readiness	Waits until the Busy bit in the Command Register is clear	Clears the Busy bit
Download tertiary module	Issues the commands to download the tertiary module into the PRISM MAC.	
Initialization	Issues the command to initialize the PRISM MAC.	Firmware carves up available NIC memory into a linked list of 128-byte blocks identified by pointers known as Frame Identifiers (FIDs), and enters the Disabled state
Wait for command completion	Waits until the Busy bit in the Command Register is clear	Clears the Busy bit
Configure the PRISM MAC for AP operation	Sets the SSID name by writing to RID 0xFC04 Waits for the Busy bit to clear before proceeding	Receives the command and updates the configuration item Clears the Busy bit
	Sets the desired default channel by writing to RID 0xFC03 Waits for the Busy bit to clear before proceeding	Receives the command and updates the configuration item Clears the Busy bit
	Sets the basic data rates by writing to RID 0xFCB3 Waits for the Busy bit to clear before proceeding	Receives the command and updates the configuration item Clears the Busy bit
	Sets the supported data rates by writing to RID 0xFCB4 Waits for the Busy bit to clear before proceeding	Receives the command and updates the configuration item Clears the Busy bit
	Sets the transmission data rate for transmissions on MAC port 0 by writing to RID 0xFC9E Waits for the Busy bit to clear before proceeding	Receives the command and updates the configuration item Clears the Busy bit
	Sets other desired configuration items by writing to the appropriate RIDs Waits for the Busy bit to clear before proceeding	Receives commands and updates configuration items Clears the Busy bit
Establish event handler(s)	Sets the desired bit(s) in the Interrupt Enable register to map processing of the Events to the appropriate interrupt handling function(s). **NOTE**: The host must acknowledge all events either through interrupts or polling, including received frames and unsolicited information frames, as soon as the information contained therein has been used or recorded by the host.	
Enable operation	Issues the Enable command	Firmware begins transmitting beacons at the specified interval and responds to probe requests, authenticate requests, associate requests, and other management frames as appropriate.

Since some functions of the access point specified in the 802.11 standard (see Reference [1]) are not under the control of the PRISM MAC in this mode, compliance with the standard is the responsibility of the host software.

The host-based AP functionality changes are enabled by setting the PortType to HostAP (see 4.4.2.1.1 cnfPortType (Station Only) for station firmware. Specifying a value of 6 allows the host to configure the firmware at run time to operate as an AP or as a station. A port status value of 8 (StartedHostAP, see 4.4.3.2.1 PortStatus) allows HostAP mode to be indicated when the firmware is enabled.

Implementing a host-based AP limits station firmware functionality to that of simple packet transfer and some low-level 802.11 MAC management. The remainder of the AP functionality is then implemented in host software along with bridging functions. This process requires that a developer wishing to implement an AP be familiar with the 802.11 MAC management protocol, but allows the developer full flexibility in the implementation of optional and unspecified features. See Chapter 4, Command Interfaces, for information related to driver API changes.

3.2.1 FIRMWARE FUNCTIONALITY REQUIRED FOR HOST-BASED AP SUPPORT

The basic concept for a host-based AP is that the firmware acts as a frame pipe, transmitting pre-formatted 802.11 frames from the host and passing received 802.11 frames to the host. In addition to frame transmission and reception, some 802.11 MAC management functions are implemented in the firmware due to their time-critical nature. For implementation of a host-based AP, the firmware provides the following functions.

- Frame Transmission
- Frame Reception
- Beacon/Probe Response Transmission

3.2.1.1 Frame Transmission

A host-based AP needs the ability to transmit data and management frames to destination nodes. The host must specify the frame control information (Protocol Version, Type, Subtype, To/From DS, Pwr Mgt, More Data, WEP, and Order) for each transmitted frame.

Since the AP transmits frames to a number of different nodes, as opposed to a single node as in station mode, the host must specify to the firmware the rate at which frames are to be sent to particular nodes.

The firmware determines the rate at which a frame is to be transmitted as follows. During the high-level MAC processing of the transmit frame, the firmware reads the RxRate field of the Frame Descriptor (see 4.3.2.3.2 TxRate Field (word offset 5) – Host AP only) to determine the rate (in 100 kb/s units) at which frames transmit. If the host sets the value of this field to 0, the firmware uses the rate at which a frame was last received from a node as the frame transmit rate.

Due to their time-critical nature, the firmware provides the fragmentation and retry functions required for 802.11 compliance.

In Host AP mode, updating of these fields by the firmware is fine with the exception of the Address2 field. In Host AP mode, the Address2 field is updated with the current address of the node.

3.2.1.2 Frame Reception

A host-based AP must be able to receive frames directed to it and extract the frame control information from the received frame. In addition, the host must determine the rate at which the frame was received.

3.2.1.3 Beacon/Probe Response Transmission

An access point must be able to transmit beacons at required times and to transmit responses to probe requests. These frames contain timestamp information required to have microsecond accuracy. Therefore, these frames are generated by the firmware. The information needed to create these frames is provided by the host when the BSS is started.

Beacon generation is controlled by the BcnMgt process, which is enabled when the Enable command is received and the Port Type has been configured as HostAP. Probe response generation is controlled by firmware on receipt of a valid probe request.

3.2.1.4 Management Frame Processing

In firmware-based AP mode, the firmware only passes frames of type data to the host. All other frame types and subtypes are handled in the firmware. In contrast, a host-based AP implementation requires the firmware to pass most management frames to the host.

In Host AP mode, the firmware passes all management frames, except probe request frames, up to the host. This allows host software to manage associations and authentications. (Probe request frames are not passed to the host since probe response frames are generated in the firmware and the host is not required to take any further action on the receipt of a probe request.) MessageType 4, in the Status field of a Receive descriptor (see 4.3.2.5.1 Status Field (word offset 0), is set to allow the host software to identify and process received management frames.

3.3 Inter-Access Point Protocol

An Inter-Access Point Protocol (IAPP) is an extension beyond the IEEE 802.11 standard to support interoperability, mobility, handover, and coordination between APs in a wireless LAN. It is typically implemented on top of IP, and uses UDP/IP and SNAP as the transfer protocol. For roaming to work, a developer must implement some sort of IAPP between APs.

Without an IAPP, an AP will still have information on a station in its routing tables after that station has moved to a different AP. Any frames received for that station will be sent out on the wireless medium, rather than directed to the new AP for that station.

NOTE: The implementation of an IAPP is outside the scope of the 802.11 Standard, and is considered the responsibility of the AP manufacturer. Intersil does not offer an IAPP to its customers.

An IAPP performs two functions:

- Update filter tables in intermediate MAC bridges
- Inform the old AP that it no longer has a particular station associated with it

Typically, when a station associates with a new AP, that AP uses an IAPP message to send a notification of the new station association over the backbone using the station's address as the source address of the packet. This will cause any learning bridges on the network to update their routing tables.

Also, the new AP must send an IAPP message on the backbone to the old AP to let it know that the station is now connected to a different AP. The old AP must then issue a Handover Request notification to the firmware so that the firmware can purge the station from its internal tables.

PRISM tertiary (AP) firmware recognizes a HandoverAddress notification frame (see 4.4.5.1.1 HandoverAddress (AP Only)) that the host sends to the firmware so that the firmware can clean up its internal routing tables for this station. If this does not occur, then a frame received from the RF directed at the roaming station will be sent back out on the RF, and not to the host, because the firmware doesn't know that the station has moved to a new AP.

In our firmware, the AssociationStatus info frame (see 4.4.5.4.2 AssociationStatus) includes the address of the old AP if the association was a result of a ReAssociation action. This information is made available so that an IAPP notification frame can be sent from the new AP to the old AP.

Also, our Station firmware will respond to a directed frame from an AP with which it is not currently associated by sending a Deauthenticate frame (see Reference [1]). This provides some coverage for this hole in the Standard, but cannot completely close it.

Currently, there is no standard for IAPP although there is work in either 802.11 or WECA to define one.

Chapter 4 Command Interfaces

This chapter describes the command interfaces to the PRISM MAC firmware. It contains four major sections: 4.1 Interface Registers, 4.2 Interface Commands, 4.3 Frame Formats, and 4.4 Interface-Record Formats.

4.1 Interface Registers

A host driver communicates with the PRISM MAC through a set of Interface Registers, which are I/O addresses used to transfer commands and data. This section describes these registers and their operation.

4.1.1 I/O OFFSET ASSIGNMENT

An I/O offset is assigned to each register. This offset relates to the base port value configured for the PC Card Interface Controller (PCIC) which drives the PRISM MAC. Base port value plus I/O offset gives the Port Address through which the register is accessed.

Since all registers are word (16-bits) registers, two I/O ports (8-bits each) are used for register access. If the host software exchanges 16 bits with a register indicated by a Port Address, the hardware actually transfers the lower 8 bits via the I/O port addressed by the Port Address and the higher 8 bits via the I/O Port addressed by Port Address + 1. Therefore, each register is assigned an even I/O offset, which implies that I/O offset+1 is also used to access that register.

In total, an area of 64 consecutive I/O ports is used for PRISM MAC access. This means that 32 registers can be accessed through I/O offsets ranging from 00 to 3E hex.

At some I/O offsets a register with special hardware support is located; those registers are used for specific tasks. The other registers are completely controlled by the PRISM MAC firmware.

The following assignments are made to registers with special hardware support.

I/O Offset	Special Hardware Support	Assignment
00	Busy bit control. Fast Service indication (firmware attention signal).	Command
14	Fast Service indication.	Control
16	Busy bit control. Fast Service indication.	spare
1C	Busy bit control. Fast Service indication.	Offset0
1E	Busy bit control. Fast Service indication.	Offset1
30	Event occurrence signal reset (in combination with 34). Hardware interrupt generation (in combination with 32).	EvStat
32	Hardware interrupt generation (in combination with 30).	IntEn
34	Event occurrence signal reset (in combination with 30). Fast Service indication.	EvAck
36	Data pointer increment.	Data0
38	Data pointer increment.	Data1
3A	Direct access to MAC memory space (in combination with 3C)	AuxPage
3C	Direct access to MAC memory space (in combination with 3A)	AuxOffset
3E	Direct access to MAC memory space	AuxData

I/O offsets 12, 16, 26, and 2E are not yet assigned. I/O offsets 3A and 3C combine to create the physical memory location in the MAC.

4.1.2 SOFTWARE INTERFACES

The registers are organized in the following logical groups:

- Command / Status registers
- FID Management registers
- Buffer Access Paths (BAP)s
- Event registers
- Control register
- Host Software registers
- Auxiliary Port

Table 4.1 shows all the registers, with their respective formats.

In principle, a value written into a register is not changed by the PRISM MAC and, therefore, that value can be read back by the host software. Exceptions to this are:

- Register fields assigned to reflect PRISM MAC behavior will change as described in the specific register definitions.
- If a register Busy bit is 1 (see 4.1.2.1.1 Command Register (I/O offset 00)), the other bits are undefined.
- After initialization, most of the register bits are set to 0. For the exceptions, see 2.1 Initialization.

A write attempt to the EvStat register is always ignored; no value can be written into that register.

In general, registers are readable without restriction since read actions do not influence PRISM MAC functions. An exception is read actions on the Data and AuxData registers, which influence an internal data pointer.

Values read from the following registers are always undefined:

- EvAck register
- AuxPage register
- AuxOffset register

All registers are word (16 bits) registers, hence it is assumed that the host software writes or reads 16 bits per I/O cycle.

If the hardware environment can transfer only 8 bits per I/O cycle (for example, because of a limited processor instruction set), the register word must be transferred in the following manner. First, the low-order byte (bits 0–7) should be accessed through the I/O offset (even), followed by high-order byte (bits 8–15) access through I/O offset+1 (odd). To prevent undefined PRISM MAC behavior, both bytes must be transferred before any other register is accessed.

Exceptions are the Data and AuxData registers, as described in the specific register definitions. Host software may transfer a single byte using these registers.

4.1.2.1 Command / Status Registers

These registers provide a command / status interface for control of synchronous PRISM MAC functions. The command code and related information is accepted and executed by the PRISM MAC, and then the resulting status is provided.

The command code is given in the Command register. Depending on the command code, related information is passed in specific Command register bits and, if needed, in one or more Param registers.

The resulting status is provided in specific Status register bits, together with the related command code. If also needed, result information is provided in one or more Resp registers.

Command completion and status availability is signaled and acknowledged through the Event registers.

Table 4.1: Register Formats

I/O Offset	Name	Description of bits															
		15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
		Command / Status															
00	Command	Busy	Additional info							0	0	CmdCode					
02	Param0	Command parameter 0															
04	Param1	Command parameter 1															
06	Param2	Command parameter 2															
08	Status	0	Result							0	0	CmdCode					
0A	Resp0	Command response 0															
0C	Resp1	Command response 1															
0E	Resp2	Command response 2															
		FID Management															
10	InfoFID	Information FID															
20	RxFID	Received FID															
22	AllocFID	Allocated FID															
24	TxCompl-FID	Completed transmit FID															
		BAP 0															
18	Select0	FID/RID 0															
1C	Offset0	Busy 0	Err 0	0	0	Data offset 0											0
36	Data0	Data 0															
		BAP 1															
1A	Select1	FID/RID 1															
1E	Offset1	Busy 1	Err 1	0	0	Data offset 1											0
38	Data1	Data 1															
		Event															
30	EvStat	Tick	WTErr	InfDrop	0	0	0	0	0	Info	0	DTIM	Cmd	Alloc	TxExc	Tx	Rx
32	IntEn	Tick En	WTErr En	InfDrop En	0	0	0	0	0	Info En	0	DTIM	Cmd En	Alloc En	TxExc En	Tx En	Rx En
34	EvAck	Tick Ack	WTErr Ack	InfDrop Ack	0	0	0	0	0	Info Ack	0	DTIM	Cmd Ack	Alloc Ack	TxExc Ack	Tx Ack	Rx Ack
		Control															
14	Control	AuxEn	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
		Host Software															
28	SwSupport0	Software support value 0															
2A	SwSupport1	Software support value 1															
2C	SwSupport2	Software support value 2															
		Auxiliary Port															
3A	AuxPage	Page address															
3C	AuxOffset	0	0	0	0	0	0	0	0	0	Data offset						0
3E	AuxData	Data															

A subsequent command can be issued immediately after the previous command is accepted. However, such a subsequent command is accepted only after completion of the previous command is acknowledged.

NOTE: Command-initiated processes taking place between command acceptance and status availability are considered synchronous. Other processes are referred to as asynchronous.

4.1.2.1.1 Command Register (I/O offset 00)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Busy	Additional info							0	0	CmdCode					

This register is used to write commands.

Busy is automatically set to 1 when a value is written into this register (regardless of the actual bit value written), and reset to 0 when the command is accepted. If no previous command cycle is in progress (that is, if it concerns the first command after power-up / reset, or if completion of the previous command is already acknowledged), Busy is reset within 10 microseconds. This time is extended as long as a previous command completion is not yet acknowledged.

A command value is ignored if that value is written while Busy is 1.

Details of the specific commands are described in 4.2 Interface Commands.

4.1.2.1.2 Param0-2 Registers (I/O offset 02-06)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Command parameter															

These registers are used to write the parameters for various commands.

Command parameters should be written only if the Busy bit of the Command register is 0 and before the related command is written into the Command register.

NOTE: Although a command may be written into the Command and Param registers if the Busy bit of the Command register is 0, that command is handled by the PRISM MAC only after the Cmd event of the preceding command is acknowledged.

4.1.2.1.3 Status Register (I/O offset 08)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	0	Result							0	0	CmdCode					

This register is used to read the status resulting from a command execution. CmdCode indicates the last command written to the Command Register (see 4.1.2.1.1 Command Register (I/O offset 00)). Result is one of the values listed in Table 4.4.

Status availability is signaled by the Cmd bit in the EvStat register (see 4.1.2.4.1 EvStat Register (I/O offset 30)). A following command is only accepted, and hence a subsequent status can only become available, after the current status is acknowledged with the CmdAck bit of the EvAck register (see 4.1.2.4.3 EvAck Register (I/O offset 34)).

The register value is valid only while status availability is signaled, and before that availability is acknowledged.

4.1.2.1.4 Resp0-2 Registers (I/O offset 0A-0E)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Command response															

These registers are used to read the response resulting from various command executions. The register values are valid only while status availability is signaled, and before that availability is acknowledged (see Table 4.1).

4.1.2.2 FID Management Registers

These registers provide the Frame Identifier (FID) of PRISM MAC buffers that become available as result of an asynchronous process.

A specific FID value is valid only while availability is signaled by the related event. The FID value should always be read before the related event is acknowledged.

4.1.2.2.1 InfoFID Register (I/O offset 10)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Information FID															

This register is used to read the FID of information frame structure buffers.

This FID becomes available either after Inquiry command execution, or if an asynchronous process initiated by the firmware generates unsolicited information. Information FID availability is signaled by the Info bit in the EvStat register. A subsequent Information FID can become available only after availability of the current FID is acknowledged with the InfoAck bit of the EvAck register.

Once an Information FID is acknowledged, it is no longer available to the host.

4.1.2.2.2 RxFID Register (I/O offset 20)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Received FID															

This register is used to read the FID of received frame structure buffers.

This FID becomes available as result of the asynchronous Reception process in the firmware. Received FID availability is signaled by the Rx bit in the EvStat register. A subsequent Received FID can become available only after availability of the current FID is acknowledged with the RxAck bit of the EvAck register.

Once a Received FID is acknowledged, it is no longer available to the host.

4.1.2.2.3 AllocFID Register (I/O offset 22)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Allocated FID															

This register is used to read the FID of a frame structure buffer.

This FID becomes available as result of the asynchronous Allocation process initiated by an Allocate, Transmit (with reclaim), or Notify (with reclaim) command. Allocated FID availability is signaled by the Alloc bit in the EvStat register. A subsequent Allocated FID can become available only after availability of the current FID is acknowledged with the AllocAck bit of the EvAck register.

Once an Allocated FID is acknowledged, it is assumed to be the property of the host until transmitted.

4.1.2.2.4 TxComplFID Register (I/O offset 24)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Completed transmit FID															

This register is used to read the FID of transmit frame structure buffers.

This FID may become available when the asynchronous transmission process initiated by a Transmit command is completed. Completed transmit FID availability is signaled by either the Tx bit or the TxExc bit in the EvStat register. A subsequent TxCompl FID can become available only after availability of the current FID is acknowledged with the TxAck or TxExcAck bit of the EvAck register.

4.1.2.3 Buffer Access Paths (BAPs)

These two sets of registers (BAP 0 and BAP 1) are used for PRISM MAC buffer access. The specific buffer is indicated by a FID / RID value in the Select register. The first word of the specific area within that buffer to be accessed is indicated in the Offset register.

After interpretation of those addressing values by the PRISM MAC, consecutive buffer words can be accessed by repeatedly writing or reading the Data register. It is not possible to read the same word directly after it is written; the Offset register must be written again if this is needed.

When setting up the BAPs, the Select register must be set before setting the Offset register.

NOTE: When the host needs to access another area within the same buffer, only a new Data offset value needs to be written into the Offset register; there is no need to write the FID/RID value again into the Select register in that case.

4.1.2.3.1 Select0-1 Registers (I/O offset 18 or 1A)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FID/RID															

These registers are used to write the FID/RID value that indicates the on-board buffer to be accessed.

A valid FID/RID value must be written while the Busy bit of the related Offset register is 0 and before the Data offset value is written into that Offset register. After writing this register, the Data offset must be written into the related Offset register to enable actual buffer access.

A FID value is obtained via one of the FID Management registers and can be used to indicate a specific Frame buffer.

As RID, any value in the range of FC00–FFFF hexadecimal can be used to indicate the single Record buffer (see 4.2.4.1 Access Command).

4.1.2.3.2 Offset0-1 Registers (I/O offset 1C or 1E)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Busy	Err	0	0	Data offset											0

These registers are used to write the initial buffer offset and to read the buffer access status.

A Data offset value also must be written while the Busy bit is 0 and after the FID/RID value is written into the related Select register.

The specific register fields are:

Busy: Automatically set to 1 when a value is written into this register (regardless of the actual bit value written), and reset to 0 within 100 microseconds. When Busy is reset, the addressing values processing is completed and the Err field is stable.

Err: 0 (and Busy=0) indicates that the buffer can be accessed through the related Data register. 1 (and Busy=0) indicates that the given offset points outside the buffer boundary or that the FID / RID in the related Select register is incorrect.

NOTE: Data access outside the buffer boundary should not be done by a released host software version, hence reading this bit is superfluous for such a product. However, checking this bit may be helpful in a development or debug environment to validate PRISM MAC usage.

Data offset: Word offset that indicates the location of the first buffer data word that is accessed via the related Data register.

NOTE: Data offset is converted into a data pointer, which is internally maintained during buffer access via the related Data register. The internal data pointer is not reflected in the Offset register. Hence, the Offset register contains the initially written word offset, regardless of the number of data word transfers.

4.1.2.3.3 Data0-1 Registers (I/O Offset 36 or 38)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Data															

These registers are used to write or read buffer data.

The specific buffer and buffer word initially accessed is determined by values previously written to the related Select and Offset registers. When a word (16 bits) is written to or read from these registers, the internal data pointer is automatically incremented. Thus, repeated Data register access addresses consecutive buffer words.

Initial access to a Data register is allowed only if the Busy bit of the related Offset register is 0. During subsequent Data register accesses (that is, without writes to the related Select and Offset registers), the Busy remains 0. Alternating writes to and reads from these registers are allowed.

Notice that the internal data pointer is automatically incremented for writes as well as reads. For example, read-after-write for the same buffer word is not possible without rewriting its Data offset again to the related Offset register. When data must be written into separate buffer parts, it is possible to skip the words in between by means of “dummy” Data register reads, which leaves their contents unchanged, but increments the internal data pointer.

To handle subsequent data parts that may consist of an odd number of bytes, it is also allowed to transfer a byte value (8 bits). For this, byte transfer should be addressed, either to the usual Data register I/O offset or to I/O offset+1. In the first case (even I/O offset), the low-order byte (bits 0–7) of the current word is accessed and the internal data pointer remains unchanged. In the other case (odd I/O offset), the high order byte (bits 8–15) of the current word is accessed and the internal data pointer is incremented.

Due to the automatic internal data pointer increments, the Data offset must be rewritten into the related Offset register if repeated access to the same buffer word is needed.

NOTE: For the current version of the HFA3841, high-order byte transfer is not allowed. To obtain a high-order byte, the current word should be read and the low-order byte value stripped. To write a high-order byte without affecting the low-order byte of the current word, that low-order byte should first be read from the Data register (which does not increment the internal data pointer), merged with the high-order byte value, and the result should be written as a word (16 bits) into the Data register. High byte transfers will be supported in future versions of the PRISM MAC.

4.1.2.4 Event Registers

These registers are used to signal and acknowledge events. Each specific event is signaled in one of the EvStat register bits and acknowledged with the corresponding EvAck register bit.

If the corresponding IntEn register bit is set, event signaling results in a hardware interrupt condition.

4.1.2.4.1 EvStat Register (I/O offset 30)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Tick	WTErr	InfDrop	0	0	0	0	0	Info	0	DTIM	Cmd	Alloc	TxExc	Tx	Rx

This register is read to determine if one or more events have occurred.

Event occurrence is signaled to the host by the PRISM MAC setting the related register bit to 1. That signaling must be reset by writing the corresponding bit of the EvAck register. A hardware interrupt is generated upon event occurrence if the corresponding bit of the IntEn register is set to 1.

The events related to the register bits are:

Tick: Occurs at each tick of the Auxiliary timer.

WTErr: Occurs when a Wait Time-out Error is detected by the PRISM MAC hardware.

NOTE: A Wait Time-out Error should not occur when proper firmware components are executed, hence reading this bit or related enabling for hardware interrupt generation is superfluous for a released product. However, checking this bit may be helpful in a development or debug environment; it validates that internal resources are not claimed too long by an erroneous firmware routine.

If this error occurs, further PRISM MAC behavior is unpredictable and power-up / reset should be used for error recovery.

InfDrop: Occurs when an unsolicited Frame Structure cannot be generated by an asynchronous process because there is not enough Frame Structure buffer space available (see 4.3.1.2 Frame Structure Storage Limit Handling).

Info: Occurs when an Information frame is generated by an asynchronous process and the InfoFID register contains the related Information FID.

This event occurs again if the next Information frame is already available when the current event occurrence is acknowledged.

DTIM: This AP-only event will be asserted at the beacon interval prior to the DTIM interval. This event allows the AP host to queue multicast/broadcast frames when there are stations in power-save mode.

Cmd: Occurs when command execution is completed and the Status and Response registers contain the related results.

Alloc: Occurs when a Frame Structure buffer is allocated or reclaimed, and the AllocFID register contains the FID identifying the related buffer. This event occurs again if a next Frame Structure buffer is already reclaimed when the current event occurrence is acknowledged.

TxExc: Occurs when the asynchronous transmission process is unsuccessfully completed (for example, the transmit retry limit is reached) while the TxExc bit in the TxControl field of the Transmit Frame structure is set. The TxComplFID register contains the related TxCompl FID. This event recurs if a next frame transmission exception is already detected when the current event occurrence is acknowledged.

Tx: Occurs when the asynchronous Transmission process is successfully completed while the TxOK bit in the TxControl field of the Transmit Frame structure is set. The TxComplFID register contains the related Completed transmit FID. This event recurs if a next frame transmission is already completed when the current event occurrence is acknowledged.

Rx: Occurs when a frame is received by the asynchronous Reception process. The RxFID register contains the related Received FID. This event occurs again if a next received frame is already available when the current event occurrence is acknowledged.

4.1.2.4.2 IntEn Register (I/O offset 32)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Tick En	WTErr En	InfDrop En	0	0	0	0	0	InfoEn	0	DTIM En	Cmd En	Alloc En	TxExc En	TxEn	RxEn

This register is used to write a control mask for hardware interrupt generation by events. For each mask bit that is set to 1, the hardware interrupt condition is generated when the corresponding bit (same position) of the EvStat register becomes 1.

4.1.2.4.3 EvAck Register (I/O offset 34)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Tick Ack	WTErr Ack	InfDrop Ack	0	0	0	0	0	Info Ack	0	DTIM Ack	Cmd Ack	Alloc Ack	TxExc Ack	Tx Ack	Rx Ack

This register is used to write event acknowledgments.

For each bit that is set to 1, the event signaled by the corresponding bit (same position) of the EvStat register is acknowledged. An event occurrence must always be acknowledged. Values read from this register are undefined.

Acknowledgment bits are internally set to 0 by the PRISM MAC upon acknowledgment completion. Thus, only register bits set to 1 affect PRISM MAC behavior.

Acknowledgment bits that are set to 1 while the corresponding EvStat register bit is 0 do not affect the PRISM MAC behavior, except that a related event that occurs just before this register is written may unintentionally be acknowledged without further event processing.

4.1.2.5 Control Registers

These registers are used to control or obtain information from some asynchronous PRISM MAC processes.

4.1.2.5.1 Control Register (I/O offset 14)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AuxEnable		0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register is used to write or read asynchronous control values.

Notice that a value written to a specific Control register field should not affect other register bits. Thus, for write actions, a read → modify → write sequence should always be used.

The controls related to the control value bits are:

AuxEnable: This 2-bit field controls and indicates if Auxiliary Port access is enabled. For normal PRISM MAC operation, Auxiliary Port access should be disabled. Access enabling is provided for debug, testing, and maintenance purposes.

If written, this field directs the port access control process:

1 – Auxiliary Port access disable
2 – Auxiliary Port access enable

If read, this field indicates the port access status:

0 – Auxiliary Port access is disabled
3 – Auxiliary Port access is enabled

After the port access control code is written, within 5 milliseconds the related process is completed and the resulting status indicated.

Therefore, (with a suggested protective minimal time of 5 milliseconds) after writing access disable (1) or enable (2), the AuxEnable bits should be read until they indicate that access is respectively disabled (0) or enabled (3). Only if the latter is true can the Auxiliary Port be used reliably.

To prevent accidental Auxiliary Port enabling, a specific pattern must be written into the Param0–2 registers before the access enable code (2) is written. If this pattern is not available, the port is disabled. The required pattern is:

FE01 – Param0 register
DC23 – Param1 register
BA45 – Param2 register

4.1.2.6 Host Software Registers

These registers are available for storage of host software values. The register values are neither interpreted nor changed by PRISM MAC functions, except that they are set to 0 during power-up initialization and cold initialization (see 4.2.1.1 Initialize Command).

The host software programmer can use those registers, for instance, to create a semaphore between different tasks.

4.1.2.6.1 SwSupport0-2 Registers (I/O offset 28, 2A, or 2C)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Software support value															

These registers are used to write or read a value for exchange between host software parts. A software support value is kept in the register without affecting the PRISM MAC behavior.

4.1.2.7 Auxiliary Port Registers

These registers are available for download, debug and testing purposes. They provide direct hardware access (without firmware intervention) to the PRISM MAC memory.

The memory word initially to be accessed is determined by the AuxPage and AuxOffset registers contents. Consecutive buffer words can be accessed by repeatedly writing or reading the AuxData register.

The physical address to be accessed through the AuxData register is assigned by setting bits 7–21 of the physical address in the AuxPage register and bits 0–6 in the AuxOffset register. The hardware will combine these two fields to create the physical address to which the AuxData register will point.

NOTE: After power-up / reset, Auxiliary port access is disabled, which means that all write actions to the port registers are ignored and values read from those registers are undefined.

Auxiliary Port access is (temporarily) enabled by the Download command in the course of a firmware download cycle. If Auxiliary Port access is enabled via the Control register (presumably with special development software), the port can be used for debug and testing purposes.

The Auxiliary Port gives linear access to the PRISM MAC memory. The difference from the BAP mechanism is that BAPs deal with an internal buffer structure, while the Auxiliary Port directly accesses the memory without using a buffer structure or other firmware dependent features. The Auxiliary Port can only be used properly if actual available memory and internal usage of that memory is known. These memory characteristics are determined by the hardware environment and firmware version used with the PRISM MAC.

4.1.2.7.1 AuxPage Register (I/O offset 3A)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Page address															

This register is used to write the memory page selection.

The page address is the value that indicates the memory page (128-byte memory area) in which the first word to be accessed is located.

NOTE: Values read from this register are undefined.

4.1.2.7.2 AuxOffset Register (I/O offset 3C)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	0	0	0	0	0	0	0	0	0	Data offset						0

This register is used to write the initial memory page offset.

The Data offset is a word offset that indicates where the first data word is written or read through the AuxData register.

NOTE: Data offset is placed in a data pointer, which is internally maintained during memory access via the AuxData register.

NOTE: Values read from this register are undefined.

4.1.2.7.3 AuxData Register (I/O offset 3E)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Data															

This register is used to write or read memory data. The memory word initially accessed is determined by values previously written to the AuxPage and AuxOffset registers.

When a word (16 bits) is written to or read from this register, the internal data pointer is automatically incremented. Thus, repeated AuxData register access addresses consecutive memory words.

NOTE: It is not necessary to write the next page address into the AuxPage register when the internal data pointer passes the end of the current page. In principle, the total of the consecutive memory following the first data word can be reached by successive AuxData register accesses.

Alternating writes to and reads from this register are allowed.

The internal data pointer is automatically incremented for writes as well as reads. For example, read-after-write for the same memory word is not possible without writing a new value to the AuxOffset register. When data must be written into separate memory parts, it is possible to skip the words in between by means of “dummy” AuxData register reads which leaves their contents unchanged.

Byte access, as needed to handle data parts that consist of an odd number of bytes, is possible in the same way as described for the BAPs (see 4.1.2.3.3 Data0-1 Registers (I/O Offset 36 or 38)).

4.2 Interface Commands

This section describes the functional definition of the PRISM MAC Software Interface Commands.

PRISM MAC behavior is primarily controlled by commands that are consecutively issued by the Host software. Each command initiates a specific process. When that process ends, the command execution result status is returned to the Host software.

A subsequent command can be processed only after the preceding command is accepted by the PRISM MAC. Such a subsequent command is accepted and processed only after acceptance of the preceding completion status is acknowledged by the Host software. Therefore, such a command-initiated process is called a synchronous process.

Several command-driven synchronous processes initiate or terminate asynchronous processes. An asynchronous process has distinct control and completion criteria, and particular means to provide results to the Host software.

The relationship between synchronous processes and some of the asynchronous processes is further described in the specific command definitions. The commands are organized in the following logical groups:

- Controller State commands
- Buffer Management commands
- Regulate commands
- Configure commands
- Serial I/O commands

Table 4.2 shows the commands with their respective format and (hexadecimal) command code.

Table 4.2: Command Formats and Hex Code

Command	**Description of bits**															
	15	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
	Controller State															
Initialize	Busy	0	0	0	0	0	0	0	0	0	CmdCode = 00					

Table 4.2: Command Formats and Hex Code (Continued)

Enable	Busy	0	0	0	0	MACPort			0	0	CmdCode = 01
Disable	Busy	0	0	0	0	MACPort			0	0	CmdCode = 02
Diagnose	Busy	0	0	0	0	0	0	0	0	0	CmdCode = 03
	Buffer Management										
Allocate	Busy	0	0	0	0	0	0	0	0	0	CmdCode = 0A
Transmit	Busy	0	0	0	0	0	0	Recl	0	0	CmdCode = 0B
	Regulate										
Notify	Busy	0	0	0	0	0	0	Recl	0	0	CmdCode = 10
Inquire	Busy	0	0	0	0	0	0	0	0	0	CmdCode = 11
	Configure										
Access	Busy	0	0	0	0	0	0	Write	0	0	CmdCode = 21
Download	Busy	0	0	0	0	0	ProgMode		0	0	CmdCode = 22
	Serial I/O										
ReadMIF	Busy	0	0	0	MSB	DeviceSel			0	0	CmdCode = 30
WriteMIF	Busy	0	More	IFSel	MSB	DeviceSel			0	0	CmdCode = 31

The Host software issues the commands through the Command register. For some commands, parameter values are also provided via the Param0-2 registers.

At synchronous process completion, the resulting status is returned in the Status register. For some commands, response values are returned via the Resp0-2 registers as well. For details of those registers, the proper usages, and the relationships with other registers (for example, the Event registers), see 4.1 Interface Registers.

Table 4.3 shows all the commands and their usage of the Parameters and Response registers.

Table 4.3: Command Parameters and Response Registers

Command	**Parameters**			**Responses**		
	Param0	**Param1**	**Param2**	**Resp0**	**Resp1**	**Resp2**
Controller State						
Initialize	-	-	-	-	-	-
Enable	-	-	-	-	-	-
Disable	-	-	-	-	-	-
Diagnose	PatternA	PatternB	-	PatternB	PatternA	-
Buffer Management						
Allocate	Bufferlength	-	-	-	-	-
Transmit	TransmitFID	-	-	FID	-	-
Regulate						
Notify	InformationFID	-	-	-	-	-
Inquire	InfoType	-	-	-	-	-
Configure						
Access	RID	-	-	-	-	-
Download	LowAddress	HighAddress	CodeLength	-	-	-

Table 4.3: Command Parameters and Response Registers (Continued)

Serial I/O						
ReadMIF	Address	-	-	Data	-	-
WriteMIF	Address	Data	-	-	-	-

The Address value is used only for Modem Management Interface (MMI) control. The value returned in the Status register comprises a Result field and a CmdCode field. The CmdCode value is always copied from the CmdCode field of the initiating command issued through the Command register.

The Result field may contain one of the Result Codes listed in Table 4.4

Table 4.4: Result Codes

Code	Description	Related Command(s)
00	Successful	All
01	Card failure	Diagnose
05	No buffer space	Inquire
7F	Command error	All

If a Result Code other than “Successful” is returned (in other words, if the value of the Result Code field is non-zero), the Resp0 register contains either an Error Qualifier value or a zero value (the latter can be interpreted as “No Error Qualifier available”). Table lists some of the Error Qualifier codes and their meaning.

Table 4.5: Error Qualifiers

Name	Value	Meaning
ErrQual.IllCode	0x01	Invalid command code
ErrQual.IllFmt	0x02	Invalid command format
ErrQual.InvFID	0x03	Transmit attempted using an invalid FID
ErrQual.InvRID	0x04	Attempt to access an invalid RID
ErrQual.TooLarge	0x05	Requested buffer too large, DataLen too large (Allocation command)
ErrQual.NotDis	0x06	Communications not disabled when attempting to download to RAM
ErrQual.AllocBsy	0x07	Allocation command issued while another allocation still busy
ErrQual.InvMode	0x08	Invalid mode field
ErrQual.NoTx	0x09	Attempt to transmit in Monitor only Mode
ErrQual.TstErr	0x0A	Loop test or Memory test error
ErrQual.WriteOnly	0x0B	Cannot read this RID
ErrQual.NoSpace	0x0C	Not enough buffer space available (Inquire command failure)
ErrQual.NoCallCode	0x0D	Call Code == 0 (set if MAC address is all zeros on Enable command))
ErrQual.NoAP	0x10	Scan timed out: no good AP found

Table 4.5: Error Qualifiers (Continued)

Name	Value	Meaning
ErrQual.NoAss	0x11	Scan: Association incorrect
ErrQual.NoAdr	0x20	WEP key mapping error: MAC address not in table

The Host software is expected – if possible – to pass a returned Error Qualifier value to the user of the PRISM MAC-based product without further interpretation.

An error may be persistent because the user is not able to find a solution using the advisories described in the product documentation. In that case, passing the Error Qualifier to the engineering organization can be helpful to determine the cause of the problem and to give effective and efficient help to solve the problem.

The “Command Error” Result Code may be returned if the PRISM MAC is not used in accordance with the Functional Specification.

4.2.1 CONTROLLER STATE COMMANDS

These commands are used to control the internal state of the PRISM MAC.

Initialize brings the PRISM MAC to a defined initial condition, which is a specific form of the communications disabled state.

Enable brings a specific MAC Port into the Enabled state. As long as one or more MAC Ports are Enabled, the PRISM MAC is considered to be in the Communications Enabled state, during which one or more network connections are attempted to be established and maintained.

Disable brings a specific MAC Port into the Disabled state. If all MAC Ports are Disabled, the PRISM MAC is considered to be in the Communications Disabled state, which means that no network connection is active.

Diagnose brings the PRISM MAC into a temporary diagnostics state, during which a specific diagnostic process is performed. At process completion, the result is provided and the controller enters the Communications Disabled state.

The Diagnose command should be executed only if the PRISM MAC is set to the initial condition.

4.2.1.1 Initialize Command

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Busy	0	0	0	0	0	0	0	0	0	CmdCode = 00					

The Initialize command brings the PRISM MAC into the initial condition.

Command execution performs the needed initialization actions and leaves the PRISM MAC in the Communications Disabled state, which means that all MAC Ports are disabled.

During command execution volatile memory (including the initial Record values; see 4.4.1 Record Structure Formats) is also initialized, but firmware images downloaded into volatile memory (see 4.2.4.2 Download Command) remain functional and are not overwritten.

After initialization, most of the register bits are set to 0. The exceptions are:

- The Status register contains the command execution result
- The BAP 0 and BAP 1 registers are undefined
- The Auxiliary Port registers are disabled

The EvStat register may contain obsolete event occurrence information. Therefore, all outstanding events should be acknowledged after Initialize command completion. This can be done either by directly writing FFFF hexadecimal to the EvAck register, or by first reading the EvStat register value and than writing that value to the EvAck register.

Command Controls

None.

Command Replies

Successful (normal command completion).

Sequence Considerations

This command should be issued as the first command when the Busy bit becomes 0 after power-up / reset. The command may always be performed, regardless of the internal state of the PRISM MAC.

4.2.1.2 Enable Command

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Busy	0	0	0	0	MACPort			0	0	CmdCode = 01					

This command enables data communications on one of the MAC Ports. This means that the PRISM MAC either attempts to establish and maintain a network connection via that port, or initiates AP operation, depending on the firmware present.

Command execution brings the indicated MAC Port into the Enabled state, applies the Static Configuration Entities, and starts the related asynchronous processes.

If (and only if) one or more MAC Ports are in the Enabled state, the following Communications Enabled state-related events can be signaled via the EvStat register:

- The Rx event, which signals availability of a Receive Frame Structure
- The TxExc and Tx event, which may signal transmit completion (see 4.2.2.2 Transmit Command)
- The Info event, which signals availability of an Information Frame Structure

Command Controls

MACPort field	MAC Port to be enabled
0	MAC Port 0
1–6	MAC Port 1–6 (AP Functions only)

Command Replies

Successful (normal command completion).

Sequence Considerations

This command should be executed only while the indicated MAC Port is in the Disabled state (see 4.2.1.1 Initialize Command and 4.2.1.3 Disable Command).

4.2.1.3 Disable Command

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Busy	0	0	0	0	MACPort			0	0	CmdCode = 02					

This command stops data communications on one of the MAC Ports. Command execution terminates the asynchronous processes related to the Port Enabled state and brings the indicated MAC Port into the Disabled state.

All Transmit Frame Structures that are waiting for transmission via the indicated MAC Port (see 4.2.2.2 Transmit Command), are removed from the output queue (see 4.3.2.4 Transmit Frame Status).

Receive Frame Structures received via the indicated MAC Port and not yet obtained by the Host software are dropped. Notice that a Receive Frame Structure that already caused an Rx event remains available until the related event is acknowledged.

If all MAC Ports are in the Disabled state, Communications Enabled state-related events are not signaled (see 4.2.1.2 Enable Command).

Command Controls

MACPort field	MAC Port to be disabled
0	MAC Port 0
1–6	MAC Port 1–6 (AP Functions only)

Command Replies

Successful (normal command completion).

Sequence Considerations

None.

4.2.1.4 Diagnose Command

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Busy	0	0	0	0	0	0	0	0	0	CmdCode = 03					

This command diagnoses the Card functions. Command execution terminates when all tests are successfully completed or as soon as a failure is detected.

The following Card functions are diagnosed:

- *Register Interface* functions
- *PRISM MAC* functions, including internal transmit/receive loopback
- *External RAM* storage and addressing capabilities
- *Non-volatile memory* integrity through validation of firmware images (code and/or data)
- *Synthesizers* access and locking

Register Interface function diagnosis is performed through Param0 and Param1 register contents validation. The remaining tests are performed without parameters.

If the Successful Result Code is returned, the PRISM MAC is left in the Initial condition and memory contents is unaffected; otherwise the controller state and volatile memory contents are undetermined.

Command Controls

PatternA: The Param0 register value gives the PatternA value, which is used to perform the register interface test. This value should be the one's complement of the PatternB value provided in the Param1 register.

NOTE: A value is the one's complement of another value if all 1 bits in the first value are 0 bits in the second value and vice versa.

PatternB: The Param1 register value gives the PatternB value, which is used to perform the register interface test. This value should be the one's complement of the PatternA value provided in the Param0 register.

Command Replies

Result Code: The following result codes can be returned:

- Successful - Normal command completion.
- Card failure - Diagnostic error detected.

NOTE: Together with the Card failure result, an Error Qualifier is returned, which provides test failure details for root cause analysis by the product engineering organization.

PatternB: The Resp0 register value reflects the PatternB value that has been provided via the Param1 register.

PatternA: The Resp1 register value reflects the PatternA value that has been provided via the Param0 register.

Sequence Considerations

This command should be executed only while the PRISM MAC is in the initial condition. Hence, the Initialize command must be performed before this command is executed.

4.2.2 BUFFER MANAGEMENT COMMANDS

These commands are used to manipulate PRISM MAC buffers that are used to store communications messages.

Allocate starts an asynchronous process to locate or wait for a free frame structure buffer, and to reserve that buffer for Transmit Frame Structure or Notification Frame Structure storage.

Transmit delivers a Transmit Frame Structure stored in a transfer buffer to the asynchronous communications processes to send the data frame part of that structure.

4.2.2.1 Allocate Command

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Busy	0	0	0	0	0	0	0	0	0	CmdCode = 0A					

This command initiates the allocation of a frame structure buffer. Command execution starts the asynchronous Allocation process. That process runs until enough transfer buffer space is available for frame structure storage.

The Alloc bit in the EvStat register signals buffer availability. While this bit is 1, the allocated FID value that identifies the allocated buffer can be read via the AllocFID register.

Command Controls

Bufferlength: The Param0 register value gives the buffer length, which is used to determine the buffer space to be allocated. If this buffer is used for Transmit or Notify with Reclaim, the buffer length value should be equal to or greater than the number of bytes needed for the longest frame structure to be stored in the allocated buffer.

The buffer length is an integer that specifies the number of bytes of the required buffer space, which should be an even value in the range 4–2400.

NOTE: The buffer length needed for Transmit Frame Structures is the length of the Data field plus the length of the Control fields and Header Info fields.

Command Replies

Result Code: Successful (normal command completion).

Sequence Considerations

Execution of this command is not allowed while another allocation request is in progress. A preceding Allocate command should first be followed by acknowledgment of the related Alloc event before this command is executed again.

4.2.2.2 Transmit Command

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Busy	0	0	0	0	0	0	Recl	0	0	CmdCode = 0B					

This command initiates the transmission of a MAC Frame. Command execution provides the indicated frame structure buffer, which should contain a Transmit Frame Structure, to the asynchronous processes related to the MAC Port indicated in that structure. The asynchronous processes keep the Transmit Frame Structure until the MAC Frame is successfully transmitted, or until it is determined that successful transmission is not possible.

Depending on the setting of the TxEx and TxOk bits of its TxControl field, release of the Transmit Frame Structure by the asynchronous processes is signaled by either the TxExc bit or the Tx bit of the EvStat register (see 4.3.1 Frame Structures).

If one of these EvStat bits is 1, and before that event is acknowledged, the Transmit FID value of the related buffer can be read via the TxComplFID register, and the Transmit Frame Structure fields can be examined via BAP registers.

Command Controls

Recl field: If set to 1, the frame structure buffer is reclaimed before the Transmit Frame Structure is provided to the asynchronous processes. If set to 0, the Transmit Frame Structure buffer is not reclaimed.

Reclaim means that another buffer is used for the actual transmission, such that the same transmit FID value can be used for subsequent transmissions.

To allocate the required buffer space, the asynchronous Allocation process is started. The Alloc bit in the EvStat register signals when the original buffer is available again for the next Transmit Frame Structure. While the Alloc bit in the EvStat register is 1, the transmit FID value of the original buffer can be read via the AllocFID register.

If the TxExc bit or the TxOk bit in the EvStat register becomes 1 after release of the other buffer by the asynchronous processes, the transmit FID value indicates that the other buffer can be read via the TxComplFID register. The interpreted Control Fields and Header Info segments of the Transmit Frame Structure in a reclaimed buffer are unchanged, but the Data Info segment contents is undetermined.

TransmitFID: The Param0 register value gives the transmit FID. That FID value indicates the buffer that contains the Transmit Frame Structure to be provided to asynchronous processes.

Transmit Frame Structure: Command execution as well as the asynchronous processes behavior are influenced by the Transmit Frame Structure contents. For further description, see 4.3.2.3 Transmit Frame Control.

Command Replies

Result Code: Successful (normal command completion).

FID: The Resp0 register value reflects the transmit FID of the buffer provided to the asynchronous processes. Without reclaim, this is the FID value as provided in the Param0 register; otherwise, it is the FID value of the other buffer.

Sequence Considerations

It is possible to perform this command again before preceding buffers are released by the asynchronous processes. The transmit FID should always indicate a buffer which is not in use by the PRISM MAC. Therefore, a FID value should not be used for a next execution of this command when it is not yet allocated (again) as result of either an Allocate command or the reclaim action.

After buffer control is passed to the PRISM MAC by means of this command, the Host software should not change the provided Transmit Frame Structure contents.

4.2.3 REGULATE COMMANDS

These commands are used to influence the Enabled state of the PRISM MAC such that communications-related information is either gathered and provided to the Host software, or information from the Host software is accepted to adapt process behavior. See 4.3.1.1.2 Information Frame Structures Flows for a description of the data structures defined for this information exchange.

Notify influences the PRISM MAC such that the behavior of asynchronous processes is adapted to specific communications needs.

Inquire instructs the asynchronous processes to gather specific communications-related information. When all required information is available it is delivered to the Host software through the Information frame / Info event mechanism.

4.2.3.1 Notify Command

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Busy	0	0	0	0	0	0	Recl	0	0	CmdCode = 10					

This command provides a specific Information frame to the PRISM MAC. The information is used to adapt the behavior of asynchronous processes to specific communications needs.

Command execution activates an asynchronous process to handle the provided information. That process runs until all information is processed.

Command Controls

Recl field: If set to 1, the buffer is reclaimed after the Information Frame Structure values are fetched by the asynchronous processes. If set to 0, the buffer is not reclaimed.

Reclaim means that the same buffer becomes available again, such that the same FID value can be used for subsequent notifications. To indicate buffer availability the asynchronous Allocation process is used; specifically, the Alloc bit in the EvStat register signals when the buffer is available again for the next Information Frame Structure. While the Alloc bit in the EvStat register is 1, the Information FID value of the buffer can be read via the AllocFID register.

The Information Frame Structure values in a reclaimed buffer remain unchanged.

InformationFID: The Param0 register value gives the Information FID. That FID value specifies the buffer that contains the Information Frame Structure to be provided to asynchronous processes.

Information Frame Structure: Command execution, as well as the asynchronous processes behavior, are influenced by the Information Frame Structure contents. For further description, see 4.3.1 Frame Structures.

Command Replies

Result Code: Successful (normal command completion).

Sequence Considerations

This command should be executed only while the PRISM MAC is in the Communications Enabled state (see 4.2.1.2 Enable Command).

4.2.3.2 Inquire Command

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Busy	0	0	0	0	0	0	0	0	0	CmdCode = 11					

This command requests the generation of a specific Information frame. Command execution activates an asynchronous process to obtain the requested information. That process runs until the information is gathered and put into the Information frame.

The Info bit in the EvStat register signals frame availability. While this bit is 1, the Information FID value that identifies the Information Frame buffer can be read via the InfoFID register.

Command Controls

InfoType: The Param0 register gives the Information type, which indicates the information to be gathered. The provided InfoType value should be the InfoType value of one of the Inquiry frames as defined in 4.3.1 Frame Structures.

Command Replies

Result Code: The following result codes can be returned:

- Successful - Normal command completion.
- No buffer space - Not enough buffer space available to build the Information Frame.

Sequence Considerations

This command should only be executed while the PRISM MAC is in the Communications Enabled state (see 4.2.1.2 Enable Command). A subsequent execution of this command with the same InfoType value is allowed only after the preceding execution result is acknowledged.

4.2.4 CONFIGURE COMMANDS

These commands are used to manipulate PRISM MAC memory parts, which contain values that influence and/or reflect the behavior of PRISM MAC processes.

Access copies the contents of Configuration or Information records to the Record buffer (see 4.4.1 Record Structure Formats) to enable access to its contents via BAP registers (see 4.1 Interface Registers), or writes the contents of that Record buffer to a Configuration record.

4.2.4.1 Access Command

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Busy	0	0	0	0	0	0	Write	0	0	CmdCode = 21					

This command manages records for access via BAP registers. Command execution obtains the contents of the indicated Configuration or Information record and places it in the Record buffer, or copies the contents of the Record buffer to the indicated Configuration record located in volatile memory.

The Host software can obtain and/or modify the Record buffer contents via the BAP registers if a value in the range of FC00–FFFF hexadecimal is put in the Select register (see 4.1.2.3 Buffer Access Paths (BAPs)).

Command Controls

Write field: If set to 1, the contents of the Record buffer is copied to the indicated Configuration record. If set to 0, the contents of the indicated Configuration or Information record is obtained and placed in the Record buffer.

RID: The Param0 register value specifies the RID that indicates the record involved; this should be one of the RID values defined in 4.4.1 Record Structure Formats.

Notice that an Information Record RID value is only allowed if the Write field is set to 0.

Record buffer: If the Write field is set to 1, the Record buffer contains the values to be written into the indicated Configuration record.

Command Replies

Result Code: Successful (normal command completion).

Record buffer: If the Write field is set to 0, the Record buffer contains the values obtained from the indicated record.

Sequence Considerations

To obtain the contents of a Configuration record or an Information record, the normal sequence is to perform this command with Write set to 0, and then to read the appropriate record fields from the Record buffer via the BAP registers.

To modify a Configuration record, the normal sequence is to write the appropriate record contents into the Record buffer via the BAP registers, and then to perform this command with Write set to 1.

When partial modification of a Configuration record is needed, this command is first performed with Write set to 0. The appropriate record fields are then changed in the Record buffer via the BAP registers, after which this command is performed again for the same record with Write set to 1.

If a RID value is specified that is within the RID range (FC00–FFFF hexadecimal) but for which no record is available, a special record structure is put into the Record buffer if Write is set to 0. That special structure has a zero RecordLen field value, followed by a RID field that contains the requested RID value, and of course no Data fields (see also 4.4.1 Record Structure Formats).

Specification of such an unavailable RID value with Write set to 1 is invalid. This can be prevented by reading a Configuration record and checking data availability before such a Configuration record is actually written.

4.2.4.2 Download Command

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Busy	0	0	0		0	ProgMode		0	0	CmdCode = 22					

This command controls the PRISM MAC firmware download process. During the command execution the PRISM MAC is directed to place a firmware image (code and/or data) into either volatile or non-volatile memory.

The maximum response time for Mode 3 (non-volatile memory programming) of this command is defined by the MaxLoadTime value that can be obtained via an Information Record (RID FD00, see 4.4.3.1.1 MaxLoadTime).

Notice that the Auxiliary Port should be enabled before downloading is started, and disabled after the firmware download is completed. This should be done via the Control register (see 4.1.2.5.1 Control Register (I/O offset 14)).

Command Controls

ProgMode field: The programming mode value indicates the download action to be performed:

Mode	Description
0	Disable programming
1	Enable volatile memory programming
2	Enable non-volatile memory programming
3	Program non-volatile memory section

Mode = 0 makes the downloaded firmware operational.

Mode 1 enables writing a firmware image via the Auxiliary Port registers directly into the destination area in volatile memory, starting at the provided destination address.

Mode 2 enables writing firmware image parts via the Auxiliary Port registers into a temporary download buffer in volatile memory. The starting address and length of that download buffer are defined in the DownloadBuffer record (RID FD01, see 4.4.3.1.2 DownloadBuffer).

Mode 3 writes the download buffer contents into the destination area in non-volatile memory. That area starts at the destination address provided with the preceding Enable action. The length value provided with that Enable action defines the number of bytes from the beginning of the download buffer that are written into the destination area.

LowAddress: For the Enable actions (Mode = 1 or Mode = 2), the Param0 register value gives the low-order 16 bits of the destination address. The destination address should be an even value.

Notice that the provided firmware image is handled as data during the write process, and hence the destination address is interpreted as “data space” address.

HighAddress: For the Enable actions (Mode = 1 or Mode = 2), the Param1 register value gives the high-order bits of the destination address.

NOTE: In most cases, this destination address value is also written to the AuxPage and AuxOffset registers. Be aware, however, that the format differs: the destination address value in the parameters is interpreted as a 32-bit hexadecimal value, while the Auxiliary Port must be set up with a 16-bit Page address and a 6-bit Data offset. Therefore, at least the ten most-significant bits of the HighAddress value should be zero.

CodeLength: For the Enable non-volatile memory programming action (Mode = 2), the Param2 register value defines the number of bytes to be copied from the download buffer into the destination area. This length should be an even value.

Command Replies

Result Code: Successful (normal command completion).

Sequence Considerations

The first firmware download cycle should be preceded directly by execution of the Initialize command and the enabling of the Auxiliary Port via the Control register. That first cycle may be followed by subsequent download cycles. This command, with programming mode set to enable volatile memory programming (Mode = 1), starts a volatile memory download cycle.

Then the firmware image should be written directly into the destination area via the Auxiliary Port registers. This command, with programming mode set to disable programming (Mode = 0), should be issued to complete the volatile memory download cycle and to start program execution at the destination address provided with the enable volatile memory programming action.

This command, with programming mode set to enable non-volatile memory programming (Mode = 2), starts the non-volatile memory download cycle and indicates the number of bytes that are to be written. Beware that this number should be equal to or lower than the download buffer length as provided in the DownloadBuffer record.

Then the indicated number of bytes of the firmware image should be written into the temporary download buffer via the Auxiliary Port registers starting at the address provided in the DownloadBuffer record, after which this command should be executed, with programming mode set to program non-volatile memory section, to program the provided firmware image part into non-volatile memory (Mode = 3).

If not all bytes can be written at once (for example, when the number of bytes to be downloaded is greater than the download buffer length specified in the DownloadBuffer record), this should be repeated until all bytes are written. So, a next enable action (Mode = 2) with the proper destination address and length must be executed, the next part of the firmware image must be written into the download buffer at the start address provided in the DownloadBuffer record, and then the program action (Mode = 3) must be performed again.

This command, with programming mode set to Disable programming (Mode = 0), should be used to complete the non-volatile memory programming cycle.

During the firmware download (that is, during or between the download cycles), no other commands should be executed and no BAP access should be performed. After all firmware image parts are downloaded, the Auxiliary Port should be disabled via the Control register and the Initialize command should be executed before any other command.

4.2.5 SERIAL I/O COMMANDS

These commands are used to control the Management Interface signals of the PRISM MAC. The Management Interface consists of an independent serial I/O hardware interface called the Modem Management Interface (MMI)

Although the MMI signal can be controlled by means of these commands, the effects of command execution depend on the characteristics of the connected devices. Furthermore, this signal is also directly controlled by the PRISM MAC firmware. For proper use of these commands, detailed knowledge of the connected devices and the active firmware components is needed. The control characteristics of the devices are described in Reference [4].

These commands are not needed for proper MMI manipulation during normal use, but are available to generate specific test sequences for which it is required to override the MMI control actions performed by the firmware.

There are two Serial I/O commands:

- *ReadMIF* obtains a data value from a specific device
- *WriteMIF* transfers a data or control value to a specific device

4.2.5.1 ReadMIF Command

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Busy	0	0	0	MSB	DeviceSel			0	0	CmdCode = 30					

The ReadMIF command obtains a data value from a device that is connected to the serial interface.

The provided device and address information is used to generate the serial interface data and clock signals needed to obtain the data value from the indicated device. First, the 8-bit Address value is written before the 8-bit Data value is read.

Command Controls

MSB field: If set to 1, for Address as well as Data, the most-significant bit (MSB) is transferred first and the least significant bit (lsb) is transferred last. If set to 0, the least-significant bit is transferred first and the most-significant bit is transferred last.

DeviceSel field: The device selection is:

Selection	Description
0	MMI device 0 – status registers

Address: The Param0 register gives the Address; the address value is provided as follows:

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	0	0	0	0	0	0	0	0	Address							

Command Replies

Result Code: Successful (normal command completion).

Data: The Resp0 register reflects the obtained Data value, as follows:

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	0	0	0	0	0	0	0	0	Data							

Sequence Considerations

None.

4.2.5.2 WriteMIF Command

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Busy	0	More	IFSel	MSB	DeviceSel			0	0	CmdCode = 31					

The WriteMIF command writes a data or control value to a device connected to a serial interface.

The provided interface, device, and address information is used to generate the serial interface data and clock signals needed to transfer the data value to the indicated device. On the MMI, an 8-bit Address value is written before the 8-bit Data value is transferred.

Command Controls

More field: This field is not used for the MMI serial interface.

If set to 1, the transfer cycle is not terminated after the data bits are transferred. This makes it possible to perform another WriteMIF command to transfer additional bytes if the data value destined for the connected device consists of more than 8 bits.

If set to 0, the transfer cycle is terminated after the 8 data bits are written.

IFSel field: If set to 1, not used. If set to 0, MMI is used.

MSB field: If set to 1, for Address as well as Data, the MSB is transferred first and the lsb is transferred last. If set to 0, the lsb is transferred first and the MSB is transferred last.

DeviceSel field: If MMI selected as serial interface, the device selection is:

Selection	Description
0	MMI device 0 – PRISM MAC control registers

Address: If MMI selected as serial interface, the Param0 register value gives the Address. If 1 is selected, the Param0 register content is ignored. This address value is provided as follows:

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	0	0	0	0	0	0	0	0	Address							

Data: The Param1 register gives the Data value to be transferred, as follows:

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	0	0	0	0	0	0	0	0	Data							

Command Replies

Result Code: Successful (normal command completion).

Sequence Considerations

If the selected device requires values consisting of a multiple of 8 bits, these values can be transferred by means of repeated command executions if the More field is set to 1. Only at transfer of the last group of 8 bits should the More field be set to 0.

4.3 Frame Formats

The two major types of data exchanged between the Host software and the PRISM MAC are:

- Communications data, which are transmitted and received over the wireless medium through the use of the Tx command and response to Rx events
- Local management data, which are communicated between the Host software and the PRISM MAC to denote status or configuration changes, either during operation or at initialization time, through the use of the Access, Notify, and Inquire commands or through unsolicited Info events

The local management data consist of data formatted as frames (described in this section) and data formatted as other record types that represent configuration and operational information items (described in 4.4 Interface-Record Formats).

4.3.1 FRAME STRUCTURES

Frame Structures are used to exchange buffered data structures between the Host software and the PRISM MAC.

There are two groups of frame structures:

- Communications Frame Structures
- Information Frame Structures

4.3.1.1 Frame Structure Exchange Flows

The general flows of typical Host software sequences for exchange of frames are described below.

Additional detailed descriptions of register values and command delivery are found in 4.2.2.2 Transmit Command and 4.2.4.1 Access Command. Note that events such as busy checking and event handling are not included in these flows.

4.3.1.1.1 Communications Frame Structure Flows

Transmit messages are transferred from the Host software to the PRISM MAC by means of a Transmit Frame Structure. This sequence is represented in Table 4.6.

Table 4.6: Transmit Frame Exchange

Sequence	Notes
Put Datalen in Param0 register	Use longest expected frame length if used for Transmit with Reclaim.

Table 4.6: Transmit Frame Exchange (Continued)

Put Allocate in Command register	The Alloc event signals Allocated FID availability.
Get Allocated FID from AllocFID register	Allocated FID availability may also be the result of a Transmit with Reclaim command.
Put Allocated FID in Select0 register	Alternatively, the Select1 register may be used.
Put Transmit Frame Structure in buffer using the Offset0 and Data0 registers	If Select1 was used in the previous step, the Offset1 and Data1 registers should be used.
Put TransmitFID in Param0 register	The TransmitFID value is the same as the Allocated FID value.
Put Transmit in Command register	This places the Transmit Frame Structure in the output queue.
Get Completed transmit FID from TxCmplFID register	Either the TxEx or TxOk event only signals Transmit completion if it has been indicated in the frame structure that post-process access is required (e.g., for diagnostics).
Put Completed transmit FID in Select0 register	Alternatively, the Select1 register may be used.
Get Transmit Frame Structure status information from buffer using the Offset0 and Data0 registers	If Select1 was used in the previous step, the Offset1 and Data1 registers should be used.

Received messages are transferred from the PRISM MAC to the Host software by means of a Receive Frame Structure. This sequence is represented in Table 4.7.

Table 4.7: Receive Frame Exchange

Sequence	Notes
Get Received FID from RxFID register	The Rx event signals Received FID availability.
Put FID in Select0 register	Alternatively, the Select1 register may be used.
Get Receive Frame Structure from buffer using the Offset0 and Data0 registers	If Select1 was used in the previous step, the Offset1 and Data1 registers should be used.

4.3.1.1.2 Information Frame Structures Flows

The Host software notifies the PRISM MAC about communications-related information by means of a Notification Frame. This sequence is represented in Table 4.8.

Table 4.8: Notification Frame Exchange

Sequence	Notes
Put Datalen in Param0 register	Use longest expected frame length if used for Notify with Reclaim.
Put Allocate in Command register	The Alloc event signals Allocated FID availability.
Get FID from AllocFID register	Allocated FID availability may also be the result of a Notify with Reclaim command.
Put FID in Select0 register	Alternatively, the Select1 register may be used.
Put Notification Frame in buffer using the Offset0 and Data0 registers	If Select1 was used in the previous step, the Offset1 and Data1 registers should be used.
Put InformationFID in Param0 register	The InformationFID value is the same as the Allocated FID value.
Put Notify in Command register	This activates Notification Frame handling.

The PRISM MAC provides solicited communications-related information to the Host software by means of an Inquiry Frame. This sequence is represented in Table 4.9.

Table 4.9: Inquiry Frame Exchange

Sequence	Notes
Put InfoType in Param0 register	Specifies the information to be obtained.
Put Inquire in Command register	This starts acquisition of the requested information.
Get Information FID from InfoFID register	The Info event signals Information FID availability.
Put Information FID in Select0 register	Alternatively, the Select1 register may be used.
Get Inquiry Frame from buffer using the Offset0 and Data0 registers	If Select1 was used in the previous step, the Offset1 and Data1 registers should be used.

The PRISM MAC provides unsolicited communications-related information to the Host software by means of an Unsolicited Frame. This sequence is represented in Table 4.10.

Table 4.10: Unsolicited Frame Exchange

Sequence	Notes
Get Information FID from InfoFID register	The Info event signals Information FID availability.
Put Information FID in Select0 register	Alternatively, the Select1 register may be used.
Get Unsolicited Frame from buffer using the Offset0 and Data0 registers	If Select1 was used in the previous step, the Offset1 and Data1 registers should be used.

4.3.1.2 Frame Structure Storage Limit Handling

For on-board storage of Frame Structures, one buffer pool is used.

The number of Frame Structures that can be stored depends on the actual volatile memory size, the storage needed for other administration, the availability of firmware images in volatile memory, and the actual space needed for each of the stored frames.

Under normal circumstances, timely handling of frames provided by the Host software and timely obtaining of frames to be passed to the Host software should prevent the exhaustion of the buffer pool. If under exceptional conditions not enough buffer space is available, the following may occur.

- For Transmit Frame Structures and Notification Frame Structures, buffer space must first be allocated before the structure contents can be provided to the PRISM MAC. The asynchronous allocation process (initiated by either the Allocate command or by the Transmit or Notify commands with reclaim) always runs until the requested buffer space is available. If frame handling and acquisition continues, eventually the Allocate event will signal buffer availability. That buffer remains available until frame handling is completed.
- For each received MAC Frame, a structure buffer is allocated internally by the asynchronous reception process. If not enough buffer space is available, further frame reception is skipped and the RxDiscardsNoBuffer tally is incremented (see 4.4.5.2.1 CommunicationTallies). Since the communications protocol processes should be robust for lost messages (for example, no ACK Frame is generated in this case, which causes a directed message transmission retry), incidental occurrence results only in difficult-to-notice performance degradation.
- The structure buffer for Inquiry Frames is allocated during the synchronous Inquire command process. If not enough buffer space is available, the command returns the “No buffer space” Result Code. This informs the Host software that an Info event to signal frame availability will not be generated. If frames handling and acquisition continues, the Inquire command can be repeated after a while for a next attempt to get the desired information.
- For each Unsolicited Frame, a structure buffer is allocated internally by the asynchronous process that has the information available. If not enough buffer space is available, the information is dropped and the InfDrop event is generated (see 4.1.2.4 Event Registers). This

event can be used to monitor buffer space shortage for engineering purposes, to maintain a diagnostic tally in the Host software, or to give a general warning to a user that information integrity may be negatively affected.

4.3.2 COMMUNICATIONS FRAME STRUCTURE FORMATS

Two Communications Frame Structures are used to pass information and data related to MAC Frame transfer over the RF medium between the host software and PRISM MAC:

- The *Transmit Frame Structure* provides information and data needed to construct a MAC Frame for transfer, and can also be used to obtain transfer process completion status information.
- The *Receive Frame Structure* obtains the information and data related to a received MAC Frame.

Communications Frame Structures exchanged between the Host software and the PRISM MAC consist of four main segments, as illustrated by Table 4.11.

4.3.2.1 Frame Descriptor

The first section of an allocated buffer is called the Frame Descriptor (see 1.4.6 Frame Descriptor). The Frame Descriptor, or FD, contains all needed information for transmission or reception, queuing, etc. It also contains the MAC Header, which is reused if the transmitted frame is fragmented.

Each FD is given an identification, called the FID. The FID is used to identify the correct FD in requests from the Host.

For transmission, the Control segment is used by the Host software to control how the frame is handled by the PRISM MAC, and by the PRISM MAC to inform the Host software about Transmit Frame-related status information. The Header information and Data segments are used to build the MAC Frame for data transfer in accordance with the MAC Frame Format as defined in clause 7 of Reference [1].

For reception, the Control segment is used by the PRISM MAC to inform the Host software about Received Frame-related status information. The Header information and Data segments are derived from the received MAC Frame.

Figure 4.1 shows the different Communications Frame Structure segments, and how these relate to PRISM MAC processing and the transferred MAC Frame:

Table 4.11: Communications Frame Structures Formats

Word Offset	Transmit Frame Structure		Structure Segment	Receive Frame Structure	
0	Status		Control	Status	
1	0 0		Common for	Timestamp (32-bit, 1 microsecond resolution)	
3	SwSupport		802.11	Signal	Silence
4			and		Rate
5	TxRate	RetryCount *	802.3		
6	TxControl			Status	
7	FrameControl	**	Header Info	FrameControl	
8	Duration/ID			Duration/ID	
9	Address1		802.11 Format	Address1	
12	Address2	**		Address2	
15	Address3			Address3	
18	SequenceControl	**		SequenceControl	
19	Address4			Address4	
22	DataLen			DataLen	
23	DestAddr		Header Info 802.3	DestAddr	
26	SrcAddr		Format	SrcAddr	
29	Length ***			Length ***	
30	Data [including IV (first 32 bits) and ICV (final 32 bits) if packets are encrypted by the host] Max. length – if frame is unencrypted or WEP is performed in firmware, 2304 bytes (STA) or 1500 bytes (AP). If encryption is performed by the host, max. length is 2312 bytes.		Data Info Common for 802.11 and 802.3	Data [including IV (first 32 bits) and ICV (final 32 bits) if packets are decrypted by the host] Max. length – if frame is unencrypted or WEP is performed in firmware, 2304 bytes (STA) or 1500 bytes (AP). If decryption is performed by the host, max. length is 2312 bytes.	

*** RetryCount is not implemented at present**

**** Filled in by firmware**

***** If encryption and decryption are performed by the host, IV and ICV are not indicated in the Length field**

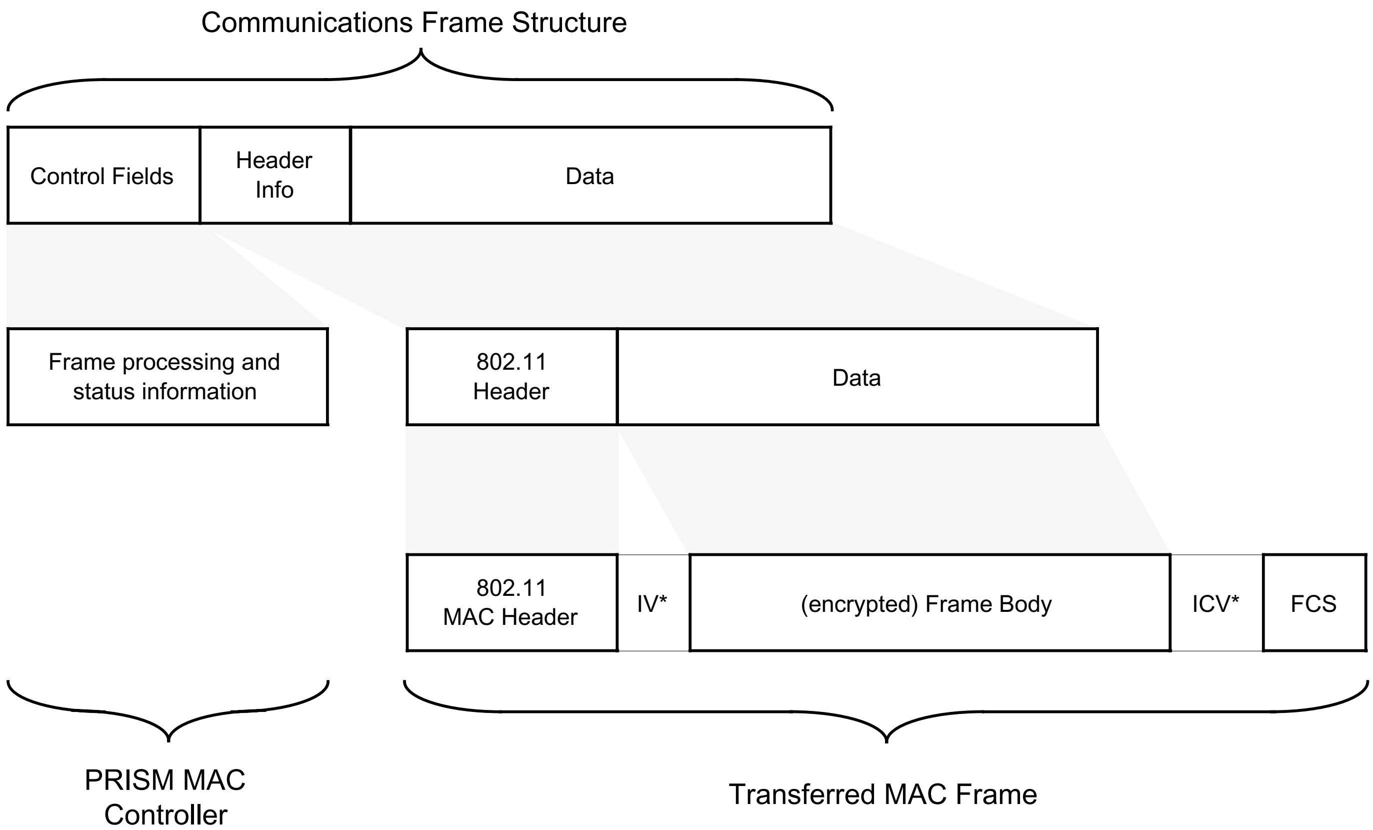


Figure 4.1: Communications Frame Structure Segments

where:

FCS = Frame Check Sequence (i.e., 32-bit Cyclic Redundancy Check).

IV= Initialization Vector, that part of the privacy key being sent in the frame.

ICV = Integrity Check Value, to check the correct data and key.

NOTE: FCS is always added (for transmission) or removed (after reception) by the PRISM MAC, and thus is not available in the Frame Structure. IV and ICV are added (for transmission) or removed (after reception) by the PRISM MAC only if the firmware is responsible for encryption and decryption (respectively).

When the host is responsible for encryption (bit 4 of cnfWEPFlags is set), IV and ICV are added by the host before the frame is given to the PRISM MAC for transmission. When the host is responsible for decryption (bit 7 of cnfWEPFlags is set), the full frame, including IV and ICV, is passed to the host after reception. See 4.4.2.2.6 cnfWEPFlags.

The Header Info in a Transmit Frame Structure contains the information in either 802.3 format or 802.11 format, as indicated by the Control fields. This information is respectively expanded into or validated as an 802.11 header by the PRISM MAC.

The Header Info in a Received Frame Structure contains the information in 802.11 format, as well as 802.3 format.

In this way, the Host Software can easily be adapted to the requirements of higher software layers. When these layers are “802.11 aware”, the Host Software uses the 802.11 format part of the Header Info. Otherwise, when those layers assume the 802.3 frame format, the PRISM MAC handles the differences between the 802.3 and the 802.11 formats without further involvement of the Host Software.

NOTE: In several Frame Structure field descriptions in the following sections, references to the IEEE 802.11 Standard [1] are made. For correct interpretation, it should be realized that most parts of the Standard describe frame fields in the sequence in which they are transferred over the RF medium, while the Frame Structures describe the internal memory representation.

4.3.2.2 Communications Frame Structure Format–Enhanced

This subsection will be defined when Retry strategy is implemented.

4.3.2.3 Transmit Frame Control

The Host software puts a Transmit Frame Structure into a PRISM MAC buffer using a transmit FID value obtained by means of the Allocate command (see 4.2 Interface Commands).

Then, using the same transmit FID value, the Transmit Frame is passed to the asynchronous Transmission process by means of the Transmit command. During Transmit command execution, the Control Fields and Header Info are interpreted to determine the frame handling and to build the actual 802.11 MAC Frame for transfer.

Notice that it is not allowed to write into the Transmit Frame Structure after it is passed to the Transmit command process.

The following sections describe how the structure fields influence Transmit command execution and the asynchronous Transmission process.

4.3.2.3.1 SwSupport Field (word offset 3)

This field is available for storage of frame-related Host software values. The field values are not interpreted for 802.11 MAC Frame transmission, and they are unchanged when the status can be obtained (see 4.3.2.4 Transmit Frame Status).

The Host software programmer can use those values, for instance, to associate a transmitted frame with other information.

4.3.2.3.2 TxRate Field (word offset 5) – Host AP only

The TxRate field gives the bitrate at which the firmware is to transmit the associated frame. The rate is specified in hexadecimal format and is used directly in the 802.11 PLCP header of the transmitted frame. Acceptable values for this field are:

Value	Description
00	Firmware controlled.
0A hexadecimal	1 Mbits/s
14 hexadecimal	2 Mbits/s
37 hexadecimal	5.5 Mbits/s
6E hexadecimal	11 Mbits/s

4.3.2.3.3 TxControl Field (word offset 6)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	0	0	0	0	0	MACPort			0	0	0	StrucType		TxEx	TxOK	0

This field contains the frame processing controls. The transmission-related processing control value fields are:

MACPort: Number of the MAC Port used to transmit the frame:

0 = MAC Port 0
1–6 = MAC Port 1–6 (AP Functions only)

Notice that the indicated MAC Port must be enabled by means of the Enable command.

StrucType: Specifies whether the 802.3 Format part or the 802.11 Format part of the Header Info segment is to be used to build the 802.11 header for transmission:

0 = 802.3 Format
1 = 802.11 Format

TxEx: Specifies whether a TxExc event is generated when 802.11 MAC Frame transmission could not successfully be completed by the asynchronous Transmission process:

0 = No TxExc event generation
1 = Generate TxExc event

TxOK : Specifies whether a TxOK event is generated when 802.11 MAC Frame transmission is successfully completed by the asynchronous Transmission process:

0 = No TxOK event generation
1 = Generate TxOK event

4.3.2.3.4 TxControl Field (word offset 6) — Enhanced

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	0	0	0	0	0	MACPort			NoEncrpt	rsrv	AltRtry	StrucType		TxEx	TxOk	0

This field contains the frame processing controls. The following-transmission related processing control value fields are added:

NoEncrpt: A value of 1 specifies that this frame is not to be encrypted.

AltRtry: A value of 1 indicates use of the alternate retry count value specified by cnfAltRetryCount (see 4.4.2.2.9 cnfAltRetryCount).

4.3.2.3.5 Header Info - 802.11 Format (word offset 7)

This 802.11 Format part of the Header Info segment is used for Transmit Frame control only if the TxControl / StrucType field is 1.

The meaning of each 802.11 Format part field is defined in clause 7 of Reference [1]. The contents of some of these 802.11 Format part fields are copied unaltered to the 802.11 header for transmission. However, the values for other fields of the 802.11 header for transmission are determined by PRISM MAC functions; therefore, the contents of the related Header Info fields is not used.

The neglected Header Info fields, which may contain any value for the Duration/ID field, the Address2 field, the Sequence Control field, and the Type, More Fragments, Retry, Power Management, and WEP sub-fields of the FrameControl field, should force the Host to set fields to 0 if not used. These fields may have future use.

The Protocol Version, Subtype, To DS, From DS, and More Data sub-fields of the Frame Control field are copied to the 802.11 header for transmission.

NOTE: It is the responsibility of the Host software to ensure that contents of the used 802.11 Format part fields comply with clause 7 of IEEE Std 802.11-1997. For example, it is specified that the Protocol Version value shall be 0; a non-zero value may cause unintended results at a receiving station.

Data provided with the 802.11 Format is always transmitted as a Data frame (Type = 10), as defined by section 7.1.3.1.2. of Reference [1]. No Host software-provided data can be transmitted as a MAC Frame with FrameControl/Type field values 00 (Management), 01 (Control), or 11 (Reserved), except when specifying 802.11 formats.

The Control and Management frames needed for proper communications are generated by PRISM MAC functions.

The Address1 and Address3 fields are always copied to the 802.11 header for transmission. For Address2, the OwnMACAddress value is always used (see 4.4.2 Configuration Records). The Address4 field is only copied if the FrameControl field To DS and From DS are both set to 1 (see section 7.2.2. of Reference [1]).

Notice that the actual meaning of the Address fields depends on the To DS and From DS bits of the FrameControl field.

The DataLen field contains the actual length of the Data field. This length indicates the number of data bytes to be transmitted in the Data field of the MAC Frame, which should be either 0 or a value not greater than cnfMaxDataLength (see 4.4.2.1.8 cnfMaxDataLength).

Notice that this value does not include the MAC header, CRC, and possible WEP IV and ICV fields of the actual 802.11 MAC Frame.

The DataLen field is formatted as an integer value. This value is used to build the MAC Frame, but in accordance with the IEEE 802.11 Standard, the value itself is not transferred over the RF medium.

4.3.2.3.6 Header Info - 802.3 Format (word offset 23)

This 802.3 Format part of the Header Info segment is used for transmit frame control only if the TxControl / StrucType field is 0.

In this case, the 802.3 Format part and the Data Info part together can be handled as the middle part of a 802.3 frame. The 802.3 MAC Frame format is described in section 3 of Reference [3]. Notice that this header is 802.3 specific, and that DIX Ethernet (also called Ethernet_II or Blue-book Ethernet) is not directly supported, as this has a Type field instead of a Length field after the address fields.

The summarized meaning of the address fields is as follows:

DestAddr: The 6-byte MAC destination address of the final recipient station(s).

SrcAddr: The 6-byte MAC source address of the initiating station.

Length: A 2-byte field that contains the actual length of the Data field. This length indicates the number of data bytes to be transferred in the Data field of the MAC Frame, which should be either 0 or a value not greater than cnfMaxDataLength (see 4.4.2.1.8 cnfMaxDataLength).

Notice that this value does not include the MAC header, CRC and possible WEP IV and ICV fields of the actual 802.11 MAC Frame.

With this information, the 802.11 header for transmission is built. For FrameControl, Type = 10 and Subtype = 0000 are used (see section 7.1.3.1.2. of Reference [1]).

The DestAddr and SrcAddr fields are used in the 802.11 header as Destination Address and Source Address respectively, in accordance with the FrameControl dependent Address rules as described in section 7.2.2. of Reference [1].

All other 802.11 header fields are set according to PRISM MAC-maintained information. If (for Station Functions) the PortType is set to portBSS, a 802.11 MAC Frame can properly be transmitted only when its Source Address is equal to the internal MAC Address of the PRISM MAC that is used by the reception process to accept frames. Therefore, with this configuration the transmit process discards a frame (see FormErr described in 4.3.2.4.1 Status Field (word offset 0)) that has a SrcAddr field content that differs from the OwnMACAddress.

The Length field is formatted as a Byte Area. The two byte values of this area are first swapped by the PRISM MAC before they are validated and used as a 16-bits integer value to build the MAC Frame.

In accordance with the IEEE 802.11 Standard, the value itself is not transferred over the RF medium.

NOTE: The Length field format is determined by 802.3 protocol stacks, which provide the most-significant part of the length value in the least-significant byte. The reason for this is that section 3.2.6. of Reference [3] states that “the length field is transmitted and received with the high order octet first”.

4.3.2.3.7 Data (word offset 30)

This field contains the data to be transmitted in the Frame Body part of the actual 802.11 MAC Frame.

The least-significant byte of the first data word becomes the first byte to be transmitted; the most-significant byte of the first data word becomes the second byte, et cetera. When the DataLen value is odd, the last data byte should be stored in the least-significant byte of the last used data word.

4.3.2.4 Transmit Frame Status

When transmit completion is signaled by the asynchronous Transmission process via either the TxExc event or the Tx event (see TxEx and TxOK in 4.3.2.3.3 TxControl Field (word offset 6)), for diagnostic purposes or exception handling, the transmit status information can be obtained from the Transmit Frame Structure fields described in the following sections.

Notice that, besides the Status field and the SwSupport field, the contents of all other Transmit Frame Structure fields are undefined at transmit completion.

4.3.2.4.1 Status Field (word offset 0)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													Form Err	Dis con	Aged Err	Retry Err

This field contains the frame transmission results.

If transmit completion is signaled by the Tx event, all four least-significant bits of this field are reset to 0, which indicates successful transmission.

If signaled by the TxExc event, one of these bits is set to 1, which indicates the reason for unsuccessful transmission as follows:

FormErr: When set, indicates that the provided Transmit Frame Structure contents is invalid. Before it is processed, the Transmit Command will be validated as follows:

- If the Source Address of this Message does not match OwnMACAddress, the command is discarded, and Tally 18 = Tx Discards - Wrong SA is incremented. This does not apply to a station’s MAC port operating on a WDS link.
- If the Frame Body length exceeds MaxDataLength, the command is discarded

If the command is discarded, and the Host requires the completion status, an Invalid status is specified.

Discon: When set, indicates that the frame was removed from the output queue because the MAC Port indicated in the TxControl field is disconnected from the network.

A MAC Port in the Disabled state (see 4.2.1.1 Initialize Command and 4.2.1.3 Disable Command) is always disconnected. The disconnection of a MAC Port in the Enabled state (see 4.2.1.2 Enable Command) depends on the related asynchronous processes. MAC Ports maintaining WDS control are always connected.

For Station Functions, a MAC Port maintaining BSS control is connected if the attempt to connect to a network (that is, association with an ESS or join with an IBSS) is successful. If this results in ESS connection, disconnection can be caused by a de-authentication or disassociation message from the related Access Point. In case of an IBSS connection, disconnection is assumed if the Connection timer expires.

For AP Functions, disconnection is assumed if the destination station is not associated with the MAC Port maintaining AP control.

AgedErr: When set, indicates that the frame was removed from the output queue because its handling was not completed before MaxTransmitLifetime (see 4.4.3.2.11 MaxTransmitLifetime) expired.

RetryErr: When set, indicates that either the ShortRetryLimit (see 4.4.3.2.9 ShortRetryLimit) or LongRetryLimit (see 4.4.3.2.10 LongRetryLimit) was exceeded for the frame or one of the fragments of that frame.

Notice that contents of the other Status field bits are undefined.

4.3.2.4.2 SwSupport Field (word offset 3)

This field contains the values that were available at transmit initiation (see 4.3.2.3 Transmit Frame Control).

NOTE: If frame transmission is initiated by a Transmit command with Reclaim, the SwSupport field contents is copied from the original buffer to the buffer that is used for the actual transmission.

4.3.2.5 Receive Frame Control

The Host software obtains a Receive Frame Structure from a PRISM MAC buffer using a Receive FID value read via the RxFID register while a Rx event is signaled.

As long as the Rx event is not yet acknowledged, the Host software may access the buffer contents. This also means that, if useful, the Host software may first process and modify parts of the frame in the buffer before the buffer contents is obtained for further processing.

The following sections describe the structure fields as set by the asynchronous Reception processes. Notice that the contents of all other (reserved) Receive Frame Structure fields are undefined.

4.3.2.5.1 Status Field (word offset 0)

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MessageType			PCF		MACPort									Un Decr	FCS Err

This field contains the frame reception results.

The reception-related status value fields are as follows:

MessageType: Type of data available in the Data field. Acceptable values for this field are:

0 = Normal
1 = RFC1042 Encoded
2 = Bridge-Tunnel Encoded
4 = 802.11 Management Frame

PCF: A values of 1 indicates that the frame was received during the CF period.

MACPort: Number of the MAC Port through which the frame is received:

0 = MAC Port 0
1–6 = MAC Port 1–6 (AP Functions only)

Notice that frames can only be received through a MAC Port, which is enabled by means of the Enable command.

UnDecr: When set, the received MAC Frame indicates that the Frame Body is encrypted while the PRISM MAC is not able to perform the decryption. The check for encryption is only performed if the calculated FCS value matches the FCS value received as part of the frame, so UnDecr is always reset when FCSErr is set.

Notice that undecrypted frames are only passed to the Host software if Promiscuous mode reception is enabled; contents of the other Status field bits are undefined.

FCSErr: When set, the received MAC Frame assumes it is corrupted, because the calculated FCS value does not match the FCS value received as part of the frame. This RID allows Receive of CRC frames.

Normally, unless directed through RID FC30, FCS error frames are not passed up to the host.

For the Receive Frame Structure contents, an FCS Error means that:

- Although this content is normally only derived from received Data frames (that is, Type = 10, see section 7.1.3.1.2. of Reference [1]), since the type cannot be determined trustworthy in this case, the information can be from any type of frame
- The Control segment and the DataLen and Length fields contain reliable information
- Although filled with received information, the Header Info - 802.11 Format (except the DataLen field) and the Data Info segments contain unreliable information
- The contents of the Header Info - 802.3 Format segment fields DestAddr and SrcAddr are undefined.

4.3.2.5.2 Signal, Silence, and Rate Fields (word offset 3)

These fields contain message-specific diagnostic values that are obtained from the RF modem by the asynchronous Reception processes.

The Signal field gives the signal level at which the 802.11 MAC Frame is received by the RF modem, while the Silence field gives the silence level as measured by the RF modem just before reception of that frame by the RF modem. Both fields are in the range 1B–9A hexadecimal.

The Rate field gives the real bitrate at which the 802.11 MAC Frame is received by the RF modem. Acceptable values for this field are:

0A hexadecimal = 1 Mbits/s
14 hexadecimal = 2 Mbits/s
37 hexadecimal = 5.5 Mbits/s
6E hexadecimal = 11 Mbits/s

The bitrate is copied from the Signal Field of the PLCP Header (see section 15.2.3.3. of Reference [1]). When the received 802.11 MAC Frame has been fragmented, the values that relate to the first fragment are shown in these fields.

4.3.2.5.3 Header Info - 802.11 Format (word offset 7)

Except for the DataLen field, clause 7 of Reference [1] defines the 802.11 Format part fields of the Receive Frame Structure. These fields are copied unaltered from the received 802.11 header.

When the received 802.11 MAC Frame has been fragmented, the values related to the first fragment are shown in these fields. The 802.11 Format is only used for received Data frames (Type = 10) (see section 7.1.3.1.2. of Reference [1]). Control and Management frames are handled by PRISM MAC functions. Actual availability of some Address fields depends on the contents of the FrameControl field.

The Address1, Address2 and Address3 fields are always available in the received 802.11 header. The Address4 field is only available if To DS and From DS are both set to 1. If an Address field is not available, the contents of the related Header Info field are undefined. Notice that the actual meaning of the Address fields depends as well on the contents of the FrameControl field.

The DataLen field contains the actual length of the Data field. This length indicates the number of data bytes received in the Data field of the MAC Frame, which should be either 0 or a value not greater than cnfMaxDataLength (see 4.4.2.1.8 cnfMaxDataLength).

Notice that this value does not include the MAC header, CRC and possible WEP IV and ICV fields of the actual 802.11 MAC Frame. The DataLen field is formatted as an Integer value. This value is derived from the MAC Frame, but in accordance with the IEEE 802.11 Standard, the value itself is not transferred over the RF medium.

4.3.2.5.4 Header Info - 802.3 Format (word offset 23)

The 802.3 Format part of the Header Info segment is used to reflect some information from the received 802.11 header, such that this part and the Data Info part together can be handled as the middle part of a 802.3 frame. The 802.3 MAC Frame format is described in section 3 of Reference [3].

The summarized meaning of the address fields is:

DestAddr: The 6-byte Destination Address value from the received 802.11 header.

SrcAddr: The 6-byte Source Address value from the received 802.11 header.

Length: A 2-byte field that contains the actual length of the Data field. This length indicates the number of data bytes received in the Data field of the MAC Frame, which should be either 0 or a value not greater than cnfMaxDataLength (see 4.4.2.1.8 cnfMaxDataLength).

Notice that this value does not include the MAC header, CRC, and possible WEP IV and ICV fields of the actual 802.11 MAC Frame.

Notice also that this value depends on the FrameControl field contents from which the 802.11 header Address (Destination Address and Source Address) values are derived (see sections 7.2.2. and 7.2.3. of Reference [1]).

The Length field is formatted as a Byte Area. When the two byte values of this area are swapped, the resulting 16-bit integer value is equal to the DataLen value in the 802.11 Format part of the Header Info segment. In accordance with the IEEE 802.11 Standard, the value itself is not transferred over the RF medium.

NOTE: The Length field format is determined by 802.3 protocol stacks, which expect the most-significant part of the length value in the least-significant byte. The reason for this is that section 3.2.6. of Reference [3] states that "the length field is transmitted and received with the high-order octet first".

4.3.2.5.5 Data (word offset 30)

This field contains the data received in the Frame Body part of the 802.11 MAC Frame.

The least-significant byte of the first data word contains the first received byte, the most-significant byte of the first data word is the second received byte, etc. Hence, when the DataLen value is odd, the last data byte is stored in the least-significant byte of the last-used data word.

4.4 Interface-Record Formats

Records are used to exchange PRISM MAC behavior-related data between the Host software and the PRISM MAC. There are two types of records:

- *Configuration records* are used to influence and adapt the PRISM MAC behavior.
- *Information records* are used to obtain configuration or operation related attributes.

Records are only indirectly accessible via a single Record buffer that has a length of at most 2500 words (that is, 5000 bytes).

The contents of a specific Record can be read into the Record buffer by means of the Access command (see 4.2.4.1 Access Command). The Access command can also be used to write the Record buffer contents into a specific Configuration record.

Record contents located in the Record buffer can be accessed using the BAP registers.

4.4.1 RECORD STRUCTURE FORMATS

The general format for Record Structures exchanged between the Host software and the PRISM MAC via the Record buffer is:

Word Offset	Field name	Size (words)
0	RecordLen: 0–249	1
1	RID: **FC*nn*–FF*nn***	1
n	Data (optional)	

The RecordLen value is an integer that specifies the number of words of the remainder of the structure. The RID value is an Integer with a specific hexadecimal content.

The specific record structure descriptions in the following sections define the Data field formats by means of a Format Descriptor. Data field contents are specified as decimal values except when specifically indicated otherwise.

ASCII Code strings are enclosed by double quotes (for example, “String”).

A specific record is indicated by its Record Identification (RID) value. The contents of a specific Configuration record or Information record can be read into the Record buffer by means of the Access command. If a RID value is specified that is within the RID range (FC00–FFEF hexadecimal) but for which no record is available, a specific record contents (RecordLen = 0; RID = the value as provided in the Param0 register; no data) is put into the Record buffer.

The Access command can also be used to overwrite the total contents of a specific Configuration record with the values available in the Record buffer. Notice that it is not allowed to write into an Information record or to write with use of a RID value for which no Configuration record is available.

All Data fields are passed between the Record buffer and the specific record location.

For Access / write actions, the RID field is not used (the RID value that specifies the destination record is provided in the Param0 register).

Access / read actions return the RID value of the requested source record in the RID field to ease record contents management by the Host software.

Notice that the RecordLen field itself is not included in the length value. For each specific RecordLen field, a decimal value or value range is defined in the specific record format definitions.

4.4.1.1 Format Descriptors

To define Data field formats, the following format descriptors are used:

Table 4.12: Format Descriptors

Descriptor	Meaning
Bit Area	Word with a specific meaning for each separate bit or smaller group of bits.
Boolean	Word that contains a value of either 0000 or 0001 hexadecimal, which indicates one of the following context-dependent meanings: 0000 = False, No, Off, or Disabled 0001 = True, Yes, On, or Enabled.
Byte	8-bit area that contains an unsigned value. Value range: 0–255 (00–FF hexadecimal).
Byte Area	1 or more consecutive Words. Each word contains 2 bytes. Storage sequence: Byte 1 in the least-significant byte of the Word at the lowest memory address, Byte 2 in the most-significant byte of that word, Byte 3 in the least-significant byte of the Word at the next higher memory address, etc.
Byte String	2 or more consecutive Words. The first Word is an Integer and the remainder is a Byte Area. The Integer contains the StringLen value, which indicates the number of significant bytes (0 or more) that are consecutively stored in the Byte Area, starting with the least-significant Byte of the Word that follows the Integer. If the Integer value is odd, the last Byte is located in the least-significant byte of the Word in which it is stored.
Integer	Word that contains an unassigned value. Value range: 0–65535 (0000–FFFF hexadecimal).

Table 4.12: Format Descriptors (Continued)

Descriptor	Meaning
Integer Area	2 or more consecutive Words. Each word contains an Integer. Storage sequence: Integer 1 is in the Word at the lowest memory address, Integer 2 is in the Word at the next higher memory address, etc.
Word	16-bit area used as a Bit area, 2 bytes, a Boolean, or an Integer For serial data transfer, the least-significant bit (bit 0) is transmitted first and the most-significant bit (bit 15) last. The least-significant Byte is located at the lower memory address.

These format descriptions are used for the specific field descriptions of the Configuration Records and Information Records definitions. For these descriptions, the following also should be noticed:

- A range definition is only given with a field description if the range of allowed values is more restrictive than the Format Descriptor ranges. In accordance with the 802.11 Standard, most time values are given in Kμseconds. 1 Kμs is equal to 1.024 milliseconds.
- The expression ‘NULL’ is used if all bits of a value are 0.
- Unless otherwise specified, all values and value ranges are decimal numbers.

A Byte String may contain a variable number of significant bytes, as indicated by the StringLen value. However, the Byte String itself is a fixed-length field that includes the bytes that do not contain a significant value. As a result, a variable number of significant bytes stored in a Byte String does not influence the RecordLen value of the record of which it is part.

Values read from Configuration records and Information records are commonly within their specified range since a proper value is set up during power-up/reset and maintained during operation. An out-of-range value written into a Configuration record field may lead to undefined behavior, including an out-of-range value as a result of a read action.

Some Information Record fields are only relevant under certain conditions. Unless otherwise specified, the actual field contents is undefined if not relevant, but the field value is always within the specified range.

Some values are specified as PDR *nnnn*, where *nnnn* is a hexadecimal number. This refers to a Production Data Record with NIC-specific values. These PDRs, as well as the ‘plugging’ process that puts the actual values into the firmware component during the firmware installation process, are described Chapter 7 Production Data Area. For each of these values, a value is defined within brackets. That default value is used if that specific value is not ‘plugged’.

4.4.2 CONFIGURATION RECORDS

The contents of Configuration records are read from and written into volatile memory. For this reason, only transient configuration values can be passed to the PRISM MAC. It is not possible to preserve a configuration value change such that it remains in effect after a power-up/reset.

At power-up/reset and during Initialize command execution, the Configuration records are filled with their default values. Only if the desired PRISM MAC behavior deviates from the behavior as characterized by the default values do Configuration records need to be written. When a Configuration record is read, the default value, or if that value is changed, the last value written into that record, is returned.

Configuration records are organized into three groups, as follows:

- Network parameters, static configuration entities
- Network parameters, dynamic configuration entities
- Behavior parameters

4.4.2.1 Network Parameters, Static Configuration Entities

Static configuration entities contain values that are copied by the synchronous Enable command process. The copied values influence the behavior of Station Enabled state-related processes. As a result, changing static configuration entity values only has effect when the Disable command is issued, followed by execution of the Enable command.

Alert: A dagger symbol (†) in the RID identification section label indicates that the RID is in development and was not fully implemented at the time this document was published.

Table 4.13 lists the static configuration entities.

Table 4.13: Network Parameters, Static Configuration Entities

RID	Name	Description
FC00	cnfPortType	[STA] Connection control characteristics
FC01	cnfOwnMACAddress	[STA] MAC Address of this node
FC02	cnfDesiredSSID (Station Only)	[STA] Service Set identification for connection
FC03	cnfOwnChannel	Communications channel for BSS creation
FC04	cnfOwnSSID	IBSS creation (STA) or ESS (AP) Service Set Identification
FC05	cnfOwnATIMWindow	[STA] ATIM Window time for IBSS creation
FC06	cnfSystemScale	System Scale that specifies the AP density
FC07	cnfMaxDataLength	Maximum length of MAC Frame Body data
FC08	cnfWDSAddress	[STA] MAC Address of corresponding WDS Link node
FC0B	cnfMulticastReceive	[STA] Switch for ESS PM Multicast reception On/Off
FC0C	cnfMaxSleepDuration	[STA] Maximum sleep time for ESS PM
FC0E	cnfOwnName	Identification text for diagnostic purposes
FC10	cnfOwnDTIMPeriod (AP Only)	[AP] Beacon intervals between successive DTIMs
FC11	cnfWDSAddress1	[AP] Port 1 MAC Address of corresponding WDS Link node
FC12	cnfWDSAddress2	[AP] Port 2 MAC Address of corresponding WDS Link node
FC13	cnfWDSAddress3	[AP] Port 3 MAC Address of corresponding WDS Link node
FC14	cnfWDSAddress4	[AP] Port 4 MAC Address of corresponding WDS Link node
FC15	cnfWDSAddress5	[AP] Port 5 MAC Address of corresponding WDS Link node
FC16	cnfWDSAddress6	[AP] Port 6 MAC Address of corresponding WDS Link node
FC17	cnfMulticastPMBuffering	[AP] Switch for PM buffering of multicast messages
FC2B	cnfMaxAssociatedStations	[AP] Limit maximum number of stations allowed to associate
FC2D	cnfRoamingMode	[STA] Configure roaming behavior
FC2E	cnfHostAuthentication	[AP] Allow Host to govern authentication process
FC30	cnfRcvCrcError	Allow frames received with FCS errors to be passed to the Host
FC34	cnfAPPCFInfo	[AP] Set configuration for AP in PCF mode
FC35	cnfSTAPCFInfo	[STA] Set configuration for CF-Aware Station
FC40	cnfTIMCtrl	Add TIM Control to control setting and clearing of TIM bits.
FC42	cnfThirty2Tally	Supports 32-bit accumulated tallies
FC43	cnfEnhSecurity	[AP] Allows configuration of enhanced security items in the AP.
FC46	cnfDbmAdjust	Unit of adjustment to convert RSSI value to dBm units
FCB0	cnfShortPreamble	Configure use of short preamble with 802.11b

Table 4.13: Network Parameters, Static Configuration Entities (Continued)

RID	Name	Description
FCB1	cnfExcludeLongPreamble	Exclude 802.11b long preamble frames
FCB3	cnfBasicRates	Define the set of rates reported as basic rates
FCB4	cnfSupportedRates	Define the set of rates supported by this MAC
FCBA	cnfPassiveScanCtrl	[STA] Define time that a station listens to a passively-scanned channel

4.4.2.1.1 cnfPortType (Station Only)

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC00**	1
2	cnfPortType	1

For station firmware, cnfPortType specifies the connection control characteristics. Values of 0, 1, and 3 specify different modes of station operation. A value of 2 specifies a “repeater” function in a wireless network. A value of 6 specifies a configuration in which the station firmware functions as the transmission and reception component of an Access Point residing in the host computer (see 3.2 Host-based AP Mode).

Field Descriptions

Name: cnfPortType

Description: Defines the connection control characteristics, where:

0 = Independent Basic Service Set (IBSS, 802.11 compliant mode)
1 = Basic Service Set (BSS)
2 = Wireless Distribution System (WDS)
3 = Pseudo IBSS (not an 802.11 compatibility mode (for testing use only))
6 = Host AP

Format: Integer

Values: Range: 0–3, 6; Default: 3

Alert: At present, station firmware does not support specification of a WDS port (cnfPortType=2).

4.4.2.1.2 cnfOwnMACAddress

Word Offset	Field name	Size (words)
0	RecordLen: 4	1
1	RID: **FC01**	1
2	cnfOwnMACAddress	3

Field Descriptions

Name: cnfOwnMACAddress

Description: Defines the MAC Address that identifies this station for data transfer.

Format: Byte Area. The least-significant bit of the first byte should be 0, indicating that it is an Individual address.

Values: Default: PDR 0101- Universal MAC Address (00, 00, 00, 00, 00, 00 hexadecimal)

4.4.2.1.3 cnfDesiredSSID (Station Only)

Word Offset	Field name	Size (words)
0	RecordLen: 18	1
1	RID: **FC02**	1
2	cnfDesiredSSID	17

Field Descriptions

Name: cnfDesiredSSID

Description: Identifies the Service Set to which the station is to connect. A zero length string indicates any SSID.

Format: Byte String

Values: Range: StringLen: 0–32; Default: StringLen = 0

4.4.2.1.4 cnfOwnChannel

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC03**	1
2	cnfOwnChannel	1

Field Descriptions

Name: cnfOwnChannel

Description: Indicates the CHNL_ID number of the communications frequency channel to be used for creating a BSS.

If the provided value is 0 or the communications channel is not allowed (see 4.4.3.1.10 ChannelList), the value in the firmware remains unchanged during Enable command execution.

Format: Integer

Values: Range: 0–14; Default: PDR 0105 - Default channel number

4.4.2.1.5 cnfOwnSSID

Word Offset	Field name	Size (words)
0	RecordLen: 18	1
1	RID: **FC04**	1
2	cnfOwnSSID	17

Field Descriptions

Name: cnfOwnSSID

Description: Identifies the Service Set to be used for creating a BSS.

Format: Byte String

Values: Range: StringLen: 1–32; Default: StringLen = 21 and Bytes = “non-specified SSID !!”

4.4.2.1.6 cnfOwnATIMWindow

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC05**	1
2	cnfOwnATIMWindow	1

Field Descriptions

Name: cnfOwnATIMWindow

Description: Defines the ATIM Window time in Kμs used for creating an IBSS. If this value is NULL, no Power Management is used within the IBSS created by this station.

Format: Integer

Values: Range: 0–100; Default: 0

4.4.2.1.7 cnfSystemScale

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC06**	1
2	cnfSystemScale	1

This setting indicates the density of APs (that is, the number of APs reachable by the station) in the ESS. The firmware uses this setting to determine when to initiate a scan for an AP offering a higher quality of communications.

Field Descriptions

Name: cnfSystemScale

Description: Specifies the Access Points density in the ESS:

1 = Low density
2 = Medium density
3 = High density

Format: Integer

Values: Range: 1–3; Default: 1

The intent of this setting is to maintain a certain level of communications quality (CQ). This is related (although not directly) to the density (or number) of APs reachable by the station.

CommsQuality (see 4.4.3.2.4 CommsQuality) indicates the quality of communications within the current BSS. The firmware derives the values for this RID based on signal level and noise level measurements. During normal operation these measurements are continuously updated by the firmware, which then calculates a signal-to-noise ratio (SNR). In principle, the higher the SNR, the better your communications quality.

CQ is used by the firmware to initiate a scan; that is, when CQ drops to a certain level, the station begins to search for an AP offering better quality communications. cnfSystemScale actually tells the firmware when to initiate a scan by setting the threshold for scanning.

A value of 3 for cnfSystemScale, indicating high density, in fact causes the station to scan for an AP that will support communications at 11 Mbps whenever the station decides (based on CQ) that the AP with which it is currently associated can no longer support this rate. This relates to AP density in the sense that in an ESS with many APs, one or more APs should be in range to support this rate.

A value of 2, indicating medium density, causes the station to scan whenever it decides that the AP with which it is currently associated can no longer support communications at rates between 5.5 Mbps and 2 Mbps. A value of 1, indicating low density, causes the station to scan whenever it decides that communications at rates between 2 Mbps and 1 Mbps can no longer be sustained.

4.4.2.1.8 cnfMaxDataLength

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC07**	1
2	cnfMaxDataLength	1

Field Descriptions

Name: cnfMaxDataLength

Description: Defines the maximum number of Data bytes that can be stored in Transmit and Receive Frame Structures (as reflected in the DataLen field of these structures; see 4.3.2.3 Transmit Frame Control and 4.3.2.5 Receive Frame Control).

Format: Integer

Values: Range: 350–2304; Default: [STA] 2304 or [AP] 1500

4.4.2.1.9 cnfWDSAddress †

Alert: At present, station firmware does not support specification of a WDS port (cnfPortType=2).

Word Offset	Field name	Size (words)
0	RecordLen: 4	1
1	RID: **FC08**	1
2	cnfWDSAddress	3

Field Descriptions

Name: cnfWDSAddress

Description: Defines the MAC Address that identifies the corresponding node if cnfPortType is set for WDS.

Format: Byte Area. The least-significant bit of the first byte should be 0, which indicates that it is an Individual address.

Values: Default: 00, 00, 00, 00, 00, 00 hexadecimal

4.4.2.1.10 cnfMulticastReceive

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC0B**	1
2	cnfMulticastReceive	1

Field Descriptions

Name: cnfMulticastReceive

Description: Specifies whether the Station receives all correctly-addressed multicast MAC Frames (see 4.4.2.2.12 GroupAddresses †) if ESS Power Management is active (see 4.4.2.2.1 cnfPMEnabled). If Off, reception of these frames does not matter.

Format: Boolean

Values: Default: Yes

4.4.2.1.11 cnfMaxSleepDuration

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC0C**	1
2	cnfMaxSleepDuration	1

Field Descriptions

Name: cnfMaxSleepDuration

Description: Defines the maximum time in Kμs that the Station is allowed to spend in Doze state if ESS Power Management is active (see 4.4.2.2.1 cnfPMEnabled).

Format: Integer

Values: Range: 1–65535; Default: 1000

4.4.2.1.12 cnfOwnName

Word Offset	Field name	Size (words)
0	RecordLen: 18	1
1	RID: **FC0E**	1
2	cnfOwnName	17

Field Descriptions

Name: cnfOwnName

Description: Identifies the printable-text name that is associated with the network node. Printable means that the text (that is, the significant bytes of the Byte String) consists only of ASCII code values from the following character ranges: a–z, A–Z, 0–9, and characters: **" ' ( ) , - . / : = ? space**.

Format: Byte String

Values: Range: StringLen: 0–31; Default: StringLen = 0

4.4.2.1.13 cnfOwnDTIMPeriod (AP Only)

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC10**	1
2	cnfOwnDTIMPeriod	1

Field Descriptions

Name: cnfOwnDTIMPeriod

Description: Specifies the number of Beacon intervals between successive Delivery Traffic Identification Maps (DTIMs).

Format: Integer

Values: Range: 1–65535; Default: 1

4.4.2.1.14 cnfWDSAddress*N* †

Alert: At present, station firmware does not support specification of a WDS port (cnfPortType=2).

Word Offset	Field name	Size (words)
0	RecordLen: 4	1
1	RID: **FC11–FC16**	1
2	cnfWDSAddress*N*	3

Field Descriptions

Name: cnfWDSAddress*N*

Description: Defines the MAC Address that identifies the corresponding node of Port*N* (range: 1–6) if cnfPortType is set for WDS. Notice that the lowest RID code indicates the lowest *N* value (1), and each next higher RID code indicates a next higher *N* value.

Format: Byte Area. The least-significant bit of the first byte should be 0, which indicates that it is an Individual address.

Values: Default: 00, 00, 00, 00, 00, 00 hexadecimal

4.4.2.1.15 cnfMulticastPMBuffering (AP Only)

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC17**	1
2	cnfMulticastPMBuffering	1

Field Descriptions

Name: cnfMulticastPMBuffering

Description: Controls whether multicast MAC Frames are buffered for transmission after DTIM. If Off, multicast MAC Frames are placed directly in the output queue.

Format: Boolean

Values: Default: Yes

4.4.2.1.16 cnfMaxAssociatedStations (AP Only)

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC2B**	1
2	cnfMaxAssociatedStations	1

Field Descriptions

Name: cnfMaxAssociatedStations

Description: The AP host software may modify the value of cnfMaxAssociatedStations in order to limit the number of stations that may be associated at any given time. Whenever a station attempts to associate with the AP, the number of associated stations is compared to cnfMaxAssociatedStations. If the current number of associate stations is equal to or greater than cnfMaxAssociatedStations, the association fails with a reason code of “AP is not able to handle additional stations”.

Format: Integer

Values: Default: 250

4.4.2.1.17 cnfRoamingMode (Station Only)

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC2D**	1
2	cnfRoamingMode	1

Field Descriptions

Name: cnfRoamingMode

Description: Controls how roaming decisions are made. Acceptable values are:

1 – Roaming Decision made by firmware
2 – Roaming Decision made by host.
3 – Roaming and Scanning Disabled

When set to 1, all roaming is handled by the firmware.

When set to 2, the host will be interrupted with a ScanResults Info Frame when either the PRISM MAC scans or the host initiates a scan. The host may issue a Join Request at that time.

When set to 3, roaming is disabled and the host is responsible for initiating scans and issuing a Join Request.

Format: Integer

Values: Range: 1–3; Default: 1

4.4.2.1.18 cnfHostAuthentication (AP Only)

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC2E**	1
2	cnfHostAuthentication	1

Field Descriptions

Name: cnfHostAuthentication

Description: Informs the firmware that the host is responsible for authenticating associating stations. When set to true, the firmware issues an AuthenticationRequest notification message and the host sends an Authentication command with an authenticateStation RID (see 4.4.2.2.27 authenticateStation (AP Only)) to the firmware to complete the authentication.

Format: Boolean

Values: Default: False.

4.4.2.1.19 cnfRcvCrcError

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC30**	1
3	cnfRcvCrcError	1

Field Descriptions

Name: cnfRcvCrcError

Description: Controls whether frames received with a CRC error are reported to the host as received frames.

Bit	Control
1	receive all frames with CRC errors
2–15	Reserved (set to 0)

Format: Bitmask

Values: True or False; Default: False

4.4.2.1.20 cnfAPPcfInfo (AP) †

Word Offset	Field name	Size (words)
0	RecordLen: 5	1
1	RID: **FC34**	1
2	cnfMediumOccupancyLimit	1
3	cnfCFPPeriod	1
4	cnfCFPMaxDuration	1
5	cnfCFPFlags	1

Field Descriptions:

Name: cnfMediumOccupancyLimit

Description: Specifies the maximum time, in Kµs, that a Point Coordinator (PC) may control the usage of the wireless medium without relinquishing control long enough to allow at least one instance of Distributed Coordination Function (DCF) access to the medium.

Format: Integer

Values: 0–1000; Default is 100

Name: cnfCFPPeriod

Description: Specifies the number of DTIM intervals between the start of Contention-Free Periods (CFPs). This is also referred to as the *Contention-Free Repetition Rate* or CFPRate.

Format: Integer

Values: default: 1

Name: cnfCFPMaxDuration

Description: Specifies the maximum duration of the CFP (in Kµs) that may be generated by the Point Coordination Function (PCF).

Format: Integer

Values: default: 0

Name: cnfCFPFlags

Description: Specifies how the CFP is utilized by the AP.

Bit	Control
0	pollStations
1	reserveDuration
2-7	reserved (set to 0)

If bit 0 is set, the PC will poll stations in the poll list during the CFP. If not set, the PC will use the CFP for its outbound traffic only.

Setting bit 1 forces the PC to reserve the full CFP for its own use. If bit 1 is not set, the PC will end the CFP before cnfCFPMaxDuration expires if there is no more data to be sent.

Format: bit field

Values: default: 0

4.4.2.1.21 cnfSTAPCFInfo (STA Only) †

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC35**	1
2	cnfCFPPollLevel	1

This RID is available on station firmware which has support for Point-Coordinator functions. The station will always report that it is ‘pollable’ and cnfCFPPollLevel will set the CF capability fields to inform the AP what level of polling support the station desires.

Field Descriptions

Name: cnfCFPPollLevel

Description: Specifies the level of polling support this station desires when participating in a PC network. These values affect the CF-Capability field values of CF-Pollable and CF-Poll Request. Supported values are:

0 – Not requesting to be on poll list.
1 – Requesting to be on poll list.
2 – Requesting to never be on poll list

Format: Integer

Values: Range: 0–2; Default: 0

4.4.2.1.22 cnfTIMCtrl

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC40**	1
3	cnfTIMCtrl	1

Field Descriptions:

Name: cnfTIMCtrl

Description: Sets and clears specific bits in the Traffic Indication Map (TIM), as follows:

0x80*nn* = Set the TIM bit for AID 0x0*nn*
0x00*nn* = Clear the TIM bit for AID 0x0*nn*

Format: Integer

Values: Range for *nn*: 0x00–0x1f

4.4.2.1.23 cnfThirty2Tally

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC42**	1
2	cnfThirty2Tally	1

Field Descriptions

Name: cnfThirty2Tally

Description: Specifies whether tallies communicated in response to an Inquire command or in an unsolicited Information frame are 16-bit (default) or 32-bit quantities.

Format: Boolean

Values: Default: False (0)

4.4.2.1.24 cnfEnhSecurity (AP Only)

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC43**	1
2	cnfEnhSecurity	1

Field Descriptions

Name: cnfEnhSecurity

Description: This bitmask value specifies whether either or both of the two enhanced security options are enabled. Acceptable values are:

Bit	Control
0	Hide SSID name in Beacon frame
1	Block response to "Unspecified-SSID"
2 – 15	Reserved (set to 0)

Setting bit 0 hides the SSID in outgoing beacon frames. With this bit set, a station cannot obtain the SSID through passive scanning.

Setting bit 1 blocks probe responses to probe requests with unspecified SSIDs. With this bit set, a station cannot authenticate without the exact SSID.

Format: Bitmask

Values: 0, 1, 2, or 3 depending on desired behavior. Default is 0, no enhanced security options enabled.

4.4.2.1.25 cnfDbmAdjust

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC46**	1
2	cnfDbmAdjust	1

Field Descriptions

Name: cnfDbmAdjust

Description: This field gives the constant used to convert RSSI (Received Signal Strength Indicator) values to dBm. The value of cnfDbmAdjust is subtracted from the RSSI value to approximate dBm values.

Format: Integer

Values: Any. Default is 100.

4.4.2.1.26 cnfShortPreamble (on supported hardware)

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FCB0**	1
2	cnfPreamble	1

Field Descriptions

Name: cnfPreamble

Description: This boolean flag specifies whether frames are transmitted with the Short or Long Preamble as defined in Reference [7]. This flag has no effect if the hardware does not support the 802.11b Short Preamble.

This flag affects the Short Preamble subfield of the Capability Information Field of Beacons, Association Response, Reassociation Response and Probe Response frames transmitted by the MAC Controller.

Bit	Control
0	longPreamble
1	shortPreamble
2–15	Reserved (set to 0)

Setting both longPreamble and shortPreamble will cause management and multicast/broadcast frames to go out with long preambles and unicast frames to use short preambles. ACK frames will use the preamble setting of the frame being ACK'd.

Hardware that does not support short preamble will ignore this configuration setting.

Format: Boolean

Values: Default: longPreamble

4.4.2.1.27 cnfExcludeLongPreamble (AP Only)

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FCB1**	1
2	cnfExcludeLongPreamble	1

Field Descriptions

Name: cnfExcludeLongPreamble

Description: This boolean flag forces the AP to disallow associations from stations that do not support short preamble (as identified in their capability flags). The default is to allow all stations to associate, which carries the risk that legacy systems may associate with the BSS and then proceed to interfere with the short preamble packets.

Format: Boolean

Values: Default: FALSE

4.4.2.1.28 cnfBasicRates

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FCB3**	1
3	cnfBasicRates	1

Field Descriptions

Name: cnfBasicRates

Description: This field is a bit-map of basic data rates. Its value specifies the basic rates used and reported for this BSS by the AP. The highest rate specified is the rate that the AP uses when transmitting broadcast/multicast and management frames. Acceptable values are:

Bit	Data Rate
0	1 Mbps
1	2 Mbps
2	5.5 Mbps
3	11 Mbps

Format: Integer

Values: Range: 1–15; Default: 3 (1 and 2 Mbps)

4.4.2.1.29 cnfSupportedRates

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FCB4**	1
3	cnfSupportedRates	1

Field Descriptions

Name: cnfSupportedRates

Description: Bit-map of supported data rates. Its value specifies the data rates reported in the capability field of management frames that contain capability fields. Acceptable values are:

Bit	Data Rate
0	1 Mbps
1	2 Mbps
2	5.5 Mbps
3	11 Mbps

Format: Integer

Values: Range: 1–15; Default: 15

4.4.2.1.30 cnfPassiveScanCtrl

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FCBA**	1
3	cnfPassiveScanCtrl	1

Field Descriptions

Name: cnfPassiveScanCtrl

Description: Defines the time (in Kmicroseconds) that the station will listen to a channel that is scanned passively. If enabled, all channels indicated in Production Data Record 104 (see 7.4.10 PDR 0x0104 - Allowed Channel Set) are actively scanned, and all other channels, except channel 14, will be passively scanned. A value of 0 disables passive scanning.

Format: Integer

Values: Range: 0–65535; Default: 0

4.4.2.2 Network Parameters, Dynamic Configuration Entities

Dynamic configuration entities contain values that immediately influence the execution of Station Enabled state-related processes. There is no need to issue a Disable command for changes to the values of these entities to take effect.

Alert: A dagger symbol (†) in the RID identification section label indicates that the RID is in development and was not fully implemented at the time this document was published.

Table 4.14 lists the dynamic configuration entities.

Table 4.14: Network Parameters, Dynamic Configuration Entities

RID	Name	Description
FC09	cnfPMEnabled	[STA] Switch for ESS Power Management (PM) On/Off
FC0A	cnfPMEPS	[STA] Switch for ESS PM EPS/PS Mode
FC0D	cnfPMHoldoverDuration	[STA] Holdover time for ESS PM
FC23	cnfWEPDefaultKeyID	Default key identifier for WEP
FC24	cnfDefaultKey0	The first of four default WEP keys
FC25	cnfDefaultKey1	The second of four default WEP keys
FC26	cnfDefaultKey2	The third of four default WEP keys
FC27	cnfDefaultKey3	The fourth of four default WEP keys
FC28	cnfWEPFlags	Flags for WEP configuration
FC29	cnfWEPKeyMapping	Table for translating mapped WEP keys
FC2A	cnfAuthentication	Control of authentication process
FC32	cnfAltRetryCnt	Set an alternate retry count for specially marked frames
FC33	cnfBeaconInterval	[AP] Set interval between beacon frames generated by the AP
FC37	cnfPriorityQUsage	Configures priority queues available to TX from CP, CFP, or both in a PC BSS.
FC80	GroupAddresses	[STA] Multicast MAC Addresses for message reception
FC81	createIBSS	[STA] Configure auto port type mode
FC82	FragmentationThreshold	[STA] Fragment length for unicast message transmission
FC83	RTSThreshold	[STA] Frame length used for RTS/CTS handshake control
FC84	TxRateControl	[STA] Data rate control for message transmission

Table 4.14: Network Parameters, Dynamic Configuration Entities (Continued)

FC85	PromiscuousMode	[STA] Switch for Promiscuous mode reception On/Off
FC90	FragmentationThreshold0	[AP] Port 0 fragment length for unicast msg transmission
FC91	FragmentationThreshold1	[AP] Port 1 fragment length for unicast msg transmission
FC92	FragmentationThreshold2	[AP] Port 2 fragment length for unicast msg transmission
FC93	FragmentationThreshold3	[AP] Port 3 fragment length for unicast msg transmission
FC94	FragmentationThreshold4	[AP] Port 4 fragment length for unicast msg transmission
FC95	FragmentationThreshold5	[AP] Port 5 fragment length for unicast msg transmission
FC96	FragmentationThreshold6	[AP] Port 6 fragment length for unicast msg transmission
FC97	RTSThreshold0	[AP] Port 0 frame length for RTS/CTS handshake control
FC98	RTSThreshold1	[AP] Port 1 frame length for RTS/CTS handshake control
FC99	RTSThreshold2	[AP] Port 2 frame length for RTS/CTS handshake control
FC9A	RTSThreshold3	[AP] Port 3 frame length for RTS/CTS handshake control
FC9B	RTSThreshold4	[AP] Port 4 frame length for RTS/CTS handshake control
FC9C	RTSThreshold5	[AP] Port 5 frame length for RTS/CTS handshake control
FC9D	RTSThreshold6	[AP] Port 6 frame length for RTS/CTS handshake control
FC9E	TxRateControl0	[AP] Port 0 data rate control for message transmission
FC9F	TxRateControl1	[AP] Port 1 data rate control for message transmission
FCA0	TxRateControl2	[AP] Port 2 data rate control for message transmission
FCA1	TxRateControl3	[AP] Port 3 data rate control for message transmission
FCA2	TxRateControl4	[AP] Port 4 data rate control for message transmission
FCA3	TxRateControl5	[AP] Port 5 data rate control for message transmission
FCA4	TxRateControl6	[AP] Port 6 data rate control for message transmission
FCB5	cnfFallbackCtrl	Configure Rate Fallback behavior
FCB6	WEPKeyDisable	Disable a previously configured WEP key
FCB7	WEPKeyMapIndex	[AP] Specify mapped WEP key for transmission to station
FCB8	BroadcastKeyID	[AP] WEP key used for broadcast frames when WEP keymapping is used
FCB9	EntSecFlag	[AP] Enable Intra-BSS security Filtering
FCE1	ScanRequest	[STA] Host initiate a scan to return a Sweep Results Table
FCE2	JoinRequest	[STA] Host initiate joining a specified BSS
FCE3	authenticateStation	[AP] Host initiate authentication of a specified Station
FCE4	ChannelInfoRequest	[AP] Request for channel info from last sweep
FCE5	HostScan	[STA] Perform non-destructive sweep of available BSSIDs

4.4.2.2.1 cnfPMEnabled

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC09**	1
2	cnfPMEnabled	1

Field Descriptions *Name:* cnfPMEnabled

Description: Controls whether Power Management is active when the Station is connected to an ESS. If Disabled, no Power Management is active during ESS connection.

Format: Boolean

Values: Default: No

4.4.2.2.2 cnfPMEPS

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC0A**	1
2	cnfPMEPS	1

Field Descriptions

Name: cnfPMEPS

Description: Controls whether the Enhanced Power Save scheme is applied when ESS Power Management is active (see 4.4.2.2.1 cnfPMEnabled). If Off, the normal Power Save scheme is applied.

Format: Boolean

Values: Default: Yes

4.4.2.2.3 cnfPMHoldoverDuration

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC0D**	1
2	cnfPMHoldoverDuration	1

Field Descriptions

Name: cnfPMHoldoverDuration

Description: Defines the time in Kµs that the Station remains in Active state after MAC Frame transfer if Enhanced Power Save is active (see 4.4.2.2.2 cnfPMEPS).

Format: Integer

Values: Range: 1–1000; Default: 100

4.4.2.2.4 cnfWEPDefaultKeyID

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC23**	1
2	cnfDefaultKeyID	1

Field Descriptions

Name: cnfDefaultKeyID

Description: Determines which entry in the Default Key table to use for transmitted packets. Using an invalid key for encryption produces an error.

Format: Integer

Values: Range: 0–3

4.4.2.2.5 cnfDefaultKey0-3

Word Offset	Field name	Size (words) (64-bit)	Size (words) (128-bit)
0	RecordLen: 4 or 8	1	1
1	RID: **FC24-FC27**	1	1
2	WEP Default Key Table Entry 0-3	3	7

Field Descriptions

Name: WEP Default Key Table Entry 0–3

Description: 64-bit or 128-bit WEP Default Key Table Entry 0–3. See Reference [1] for more information on the function of the WEP Default Key Table.

Format: The 64-bit WEP Key is the first 5 octets of the 3-word field. For 128-bit WEP, the key is the first 13 octets of the 7-word field. For both formats, octets at lower offset represent earlier octets in the WEP key.

Values: Any

4.4.2.2.6 cnfWEPFlags

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC28**	1
2	cnfWEPFlags	1

Field Descriptions

Name: cnfWEPFlags

Description: Specifies whether (and by whom) data packets are encrypted before transmission and decrypted after reception. See Table 4.11 and Figure 4.1 (including the Note following the figure).

This field is a bit-mapped value. The following flags are defined in the cnfWEPFlags field:

Bit	Control
0	PrivacyInvoked
1	ExcludeUnencrypted
2 – 3	Reserved (set to 0)
4	Host Encrypt
5 – 6	IVReuse, where 00 = every frame 01 = every 10 frames 10 = every 50 frames 11 = every 100 frames
7	Host Decrypt
8-15	Reserved (set to 0)

When bit 0 is set, packets are encrypted before transmission. This encryption is performed by the firmware, unless bit 4 is set to specify host encryption.

When bit 1 is set, the firmware will discard any unencrypted data frame (one that does not have the WEP flag set in its Control field) that it receives. Otherwise, unencrypted data frames are not excluded. Received frames are decrypted by the firmware unless bit 7 is set to specify host decryption.

With bit 4 set, the host encrypts frames before passing them to the firmware for transmission. With this bit set, the firmware simply transmits frames as they are received from the host. With bit 7 set, received frames are passed as received by the firmware to the host for decryption.

Format: Bitmap

Values: Valid combinations are all bits clear, or bit 0 set with any of bits 1, 4, 5–6, and 7 set or clear for desired behavior.

4.4.2.2.7 cnfWEPKeyMapping (AP Only)

Word Offset	Field Name	Size (words) 64-bit	Size (words) 128-bit
0	RecordLen: 8 (64-bit) or 12 (128-bit)	1	1
1	RID: **FC29**	1	1
2	STA address	3	3
5	WEP Key Index	1	1
6	WEP Key-Map Key	3	7

cnfWEPKeyMapping allows two separate WEP Key-Map Keys to be specified for a specific MAC address. This technique allows the Access Point to support two different WEP Keys for each supported Station, a requirement for rapid re-keying support. This also allows the AP to manage communications with both “Key-Mapped” and “Non Key-Mapped” Stations concurrently.

After WEP Key-Mapping has been enabled (see 4.4.2.2.22 WEPKeyMapIndex (AP Only)), the AP will attempt to use the specified key value in frame exchanges that occur between the AP and a Station.

Field Descriptions

Name: RecordLen

Description: Depends on which WEP encryption scheme has been chosen.

Values: 8 for 64-bit WEP; 12 for 128-bit WEP

Name: STA Address

Description: 48-bit MAC address of the station

Format: 48-bit IEEE 802 MAC Individual address.

Values: Any

Name: WEP Key Index

Description: Specifies which of the two WEP Keys is being updated

Values: 0 or 1

Name: WEP Key-Map Key

Description: 64-bit or 128-bit WEP Key-Mapped table entry.

Format: For 64-bit WEP, the key is the first 5 octets of the 3-word field. For 128-bit WEP, the key is the first 13 octets of the 7-word field. For both formats, octets at lower offsets represent earlier octets in the WEP key.

Values: Any

WEP Key-Mapping Overview

The AP will first attempt to locate a Node Control Block (NCB) associated with the specified MAC address. If a NCB is not found, an error is returned. If a NCB is found, the specified “Key-Mapping Key” entry of the NCB is initialized to the value specified in the WEP Key-Map Key field of this RID.

The AP first determines whether WEP Key-Mapping is enabled by examining the contents of the KeyMapOn field within the NCB. If Key-Mapping is not enabled, the AP will encrypt frames using one of four Default WEP Keys (see 4.4.2.2.4 cnfWEPDefaultKeyID). If Key-Mapping is enabled, the AP encrypts the frame using one of the two Key-Mapping Keys, which are stored in the NCB using this RID. The key used is determined by the value of the WEP Key Index field (see 4.4.2.2.22 WEPKeyMapIndex (AP Only)), which is also stored in the NCB. When this is done, the Key ID value in the transmitted frame contains this Key-Mapping index value.

Before the Station can decrypt an encrypted frame which utilizes Key-Mapping, the Station's Host software must load the appropriate Key-Mapping Keys into the WEP Default Key table. When the Station then receives an encrypted frame, the Key ID within the IV portion of the frame is examined and used as an index to select the appropriate Default WEP key.

When sending an encrypted frame back to the AP, a station will continue to use one of the Default WEP Keys to encrypt the frame. The key used is determined by the value stored in RID FC23 (see 4.4.2.2.4 cnfWEPDefaultKeyID). If WEP Key-Mapping is enabled, the Default Key ID used should always be a 0 or a 1.

When the Access Point receives an encrypted frame, the NCB associated with the sending station's MAC address is examined to determine whether or not WEP Key-Mapping is enabled. If Key-Mapping is enabled, the frame's Key ID field should specify a value of zero or one, which will cause either the KeyMapKey0 or KeyMapKey1 field within the NCB to be used to decrypt the frame. If the frame's Key ID field contains any other value while Key-Mapping is enabled, the specified Key ID value is used as an index to select one of the globally defined Default WEP Key entries (2 and 3), which effectively cause Key-Mapping to be disabled. If Key-Mapping is disabled, the Key ID within the frame's IV is used as an index to select one of the four Default WEP keys.

4.4.2.2.8 cnfAuthentication

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC2A**	1
2	cnfAuthenticationMask	1

Field Descriptions

Name: cnfAuthenticationMask

Description: The following values are defined in the cnfAuthenticationMask field:

Bit	Control
0	Open System Authentication Enable
1	Shared Key Authentication Enable

When bit 0 is set, the AP will accept Open System Authentication requests from mobile stations. Otherwise, such requests are denied with a status code "Authentication Algorithm Not Supported". When bit 1 is set, the AP will accept Shared Key Authentication requests (using the configured WEP key information). Otherwise, such requests are denied with a status code "Authentication Algorithm Not Supported".

A station will only support one type of authentication, with the default being Open System. Shared Key authentication will only be used if a valid WEP key has been configured.

Format: Bitmap

Values: Reserved flags must be set to zero (Default: Open System)

NOTE: It is not valid to enable Shared Key Authentication when no valid WEP keys have been defined.

4.4.2.2.9 cnfAltRetryCount

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC32**	1
2	cnfAltRetryCount	1

Field Descriptions

Name: cnfAltRetryCount

Description: Used to set an alternate retry count mechanism for Transmit frames that have been marked as requiring the use of an alternate retry count. Submitted Transmit frames are marked via the TxControl field as using AltRetryCount by setting bit 5 (see 4.3.2.3.4 TxControl Field (word offset 6) — Enhanced for additional information).

Format: Integer

Values: Any; Default: 7

4.4.2.2.10 cnfBeaconInterval (AP)

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC33**	1
2	cnfBeaconInterval	1

Field Descriptions

Name: cnfBeaconInterval

Description: This value specifies the beacon interval duration in milliseconds.

Format: Integer

Values: Any; Default: 100 msec; Maximum: 4095 msec

4.4.2.2.11 cnfPriorityQUsage †

The cnfPriorityQUsage RID allows maximum flexibility in how the priority transmit queues are used when operating in a PC BSS. Each of the priority queues can be configured to be available to transmit during the contention period (CP), the contention-free period (CFP), or both portions of the superframe.

The queue configuration information is only used when operating in a PC BSS. When not in a PC BSS, the queues are handled as normal priority queues to avoid orphaned MSDUs.

Word Offset	Field name	Size (words)
0	RecordLen: 3	1
1	RID: **FC37**	1
2	cnfCPUsage	1
3	cnfCFPUsage	1

Field Descriptions

Name: cnfCPUsage

Description: This Bitmask specifies whether priority queues are processed during the Contention Period of the PCF superframe. Acceptable values are:

Bit	Control
0	Priority Queue 0 (Normal Queue)
1	Priority Queue 1
2	Priority Queue 2
3	Priority Queue 3
4-15	Reserved (set to 0)

Format: Bitmask

Values: True or False; Default: True (0x000F)

Name: cnfCFPUsage

Description: This Bitmask specifies whether priority queues are processed during the Contention Free Period of the PCF superframe. Acceptable values are:

Bit	Control
0	Priority Queue 0 (Normal Queue)
1	Priority Queue 1
2	Priority Queue 2
3	Priority Queue 3
4-15	Reserved (set to 0)

Format: Bitmask

Values: True or False; Default: True (0x000F)

4.4.2.2.12 GroupAddresses †

Word Offset	Field name	Size (words)
0	RecordLen: *n**3+1; n = 0–16; Default: 1 (i.e., n = 0)	1
1	RID: **FC80**	1
2	MACAddress 1 (optional)	3
***n**3-1**	MACAddress *n* (optional)	3

Field Descriptions

Name: MACAddress

Description: Each of the 0 to 16 MAC Addresses specifies a multicast address that is checked as part of the received message acceptance procedure.

Format: Byte Area

4.4.2.2.13 createIBSS

The createIBSS RID provides the auto port-type mode required by NDIS 5.1. In conjunction with cnfPortType (see 4.4.2.1.1 cnfPortType (Station Only)), the station can attempt to join an ESS and, if an ESS is not found, create an IBSS. When cnfPortType is set to 0 (IBSS mode), the firmware uses the value set by createIBSS to determine its behavior.

If a station is the only node in auto IBSS mode, the firmware will scan every 10 seconds for an infrastructure network or an IBSS network on a different channel. If there is more than one node in the IBSS network, the firmware will not scan for a different network.

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC81**	1
2	IBSSFlags	1

Field Descriptions

Name: IBSSFlags

Description: This bitmask value specifies auto-port behavior, as follows:

Value	Meaning
00	Join or create an IBSS
01	Join ESS or join/create an IBSS
10	Join IBSS only (do not create)
11	Join ESS or IBSS (do not create an IBSS)

Format: Bitmask

Values: Default is 0, auto-port mode disabled.

4.4.2.2.14 FragmentationThreshold

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC82**	1
2	FragmentationThreshold	1

Field Descriptions

Name: FragmentationThreshold

Description: Number of bytes used for the fragmentation boundary for directed messages.

Format: Integer

Values: Range: 256–2346 (even numbers only); Default: 2346

4.4.2.2.15 RTSThreshold

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC83**	1
2	RTSThreshold	1

Field Descriptions

Name: RTSThreshold

Description: Specifies the number of bytes used for the RTS/CTS handshake boundary.

Format: Integer

Values: Range: 0–3000; Default: 2432

4.4.2.2.16 TxRateControl

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC84** (STA), **FC9E-FCA4** (AP)	1
2	TxRateControl	1

Field Descriptions

Name: TxRateControl

Description: This bit-map value specifies the data rate for message transmission. The bits in this field have the following meaning:

Bit	Data Rate
0	1Mbps
1	2 Mbps
2	5.5 Mbps
3	11 Mbps

The host may select numbers from 1 through 15 in order to control the data rate fallback algorithm.

Format: Integer

Values: Range: 1–15; Default: 3 (1 and 2 Mbps)

4.4.2.2.17 PromiscuousMode

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC85**	1
2	PromiscuousMode	1

Field Descriptions

Name: PromiscuousMode

Description: Specifies whether Promiscuous mode reception is enabled. If Disabled, only messages destined for this Station are accepted.

In promiscuous mode the station will pass up to the host all data messages it receives successfully (regardless of their destination) within its current BSS, all MPDUs received with FCS errors, all undecryptable MPDUs received successfully, and received WMP LinkTest requests (these are not responded to). This mode automatically prevents Power Management putting the RF modem in DOZE state. Roaming continues to be operational; if the station performs a scan, the monitoring station may not receive all messages due to scanning other channels.

Note that this mode differs in significant aspects from Monitor Mode (see F.2.8 TMB: Monitor Mode in Appendix G).

Format: Boolean

Values: Default: No

4.4.2.2.18 FragmentationThreshold0–6

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC90–FC96**	1
2	FragmentationThreshold*N*	1

Field Descriptions

Name: FragmentationThreshold*N*

Description: Specifies the fragmentation boundary for directed messages transmitted via Port *N* (range: 0–6). Notice that the lowest RID code indicates the lowest *N* value (0), and each next higher RID code indicates a next higher *N* value.

Format: Integer

Values: Range: 256–2346 (even numbers only); Default: 2346

4.4.2.2.19 RTSThreshold0–6

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FC97–FC9D**	1
2	RTSThreshold*N*	1

Field Descriptions

Name: RTSThreshold*N*

Description: Specifies the RTS/CTS handshake boundary for transmissions via Port *N* (range: 0–6). Notice that the lowest RID code indicates the lowest *N* value (0), and each next higher RID code indicates a next higher *N* value.

Format: Integer

Values: Range: 0–3000; Default: 2432

4.4.2.2.20 cnfFallbackCtrl

Word Offset	Field Name	Size (words)
0	RecordLen: 2	1
1	RID: **FCB5**	1
2	cnfFallbackCtrl	1

cnfFallbackCtrl provides a means to control the fallback algorithm used when the station and AP are unable to communicate at the current rate. This value can be changed at any time and becomes effective on the next transmission.

Field Descriptions

Name: cnfFallbackCtrl

Description: Specifies when to fall back to a lower data rate and when to use RTS frames to announce packet data to the receiving station. A value of 1 (one) in a bit position indicates a fallback in data rate or an RTS sequence on a particular retry, as follows:

Bit	Control
0	Data rate fallback on first retry
1	Data rate fallback on second retry
2	Data rate fallback on third retry
3	Data rate fallback on fourth retry
4	Data rate fallback on fifth retry
5	Data rate fallback on sixth retry
6	Data rate fallback on seventh retry
7	Reserved (set to 0)
8	Begin RTS control on first retry
9	Begin RTS control on second retry
10	Begin RTS control on third retry
11	Begin RTS control on fourth retry
12	Begin RTS control on fifth retry
13	Begin RTS control on sixth retry
14	Begin RTS control o seventh retry
15	Reserved (set to 0)

Format: Bitmask

Values: Default: 0x047C (data rate fallback on third through seventh retries, begin RTS control on third retry)

4.4.2.2.21 WEPKeyStatus

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FCB6**	1
3	WEPKeyStatus	1

PRISM firmware maintains a bitmask with a validity bit for each WEP key configured. Writing a value to this RID sets the status of these keys; reading this RID returns the current status of these keys.

Field Descriptions

Name: WEPKeyStatus

Description: Each bit specifies the status of a WEP key. A value of 1 (one) in a bit position indicates that the specified WEP key is disabled, as follows:

Bit	Key
0	WEP key 0
1	WEP key 1
2	WEP key 2
3	WEP key 3

Format: Bitmask

Values: 0 (enabled) or 1 (disabled). Default: 0 (all WEP keys enabled)

4.4.2.2.22 WEPKeyMapIndex (AP Only)

Word Offset	Field name	Size (words)
0	RecordLen: 5	1
1	RID: **FCB7**	1
2	STA Address	3
5	WEPKeyMapIndex	1

This RID specifies whether the AP uses WEP Key-Mapping for transmitting to the specified station, and if so, which of the two WEP keys it uses. Specifying KeyMapKey0 or KeyMapKey1 automatically enables the WEP Key-Mapping feature for transmitting (see 4.4.2.2.7 cnfWEPKeyMapping (AP Only)) and stores the index in the NCB for the specified station. Specifying a value of 0xFF disables WEP Key-Mapping for transmitting to this station.

Field descriptions

Name: STA Address

Description: 48-bit MAC address of the station

Format: 48-bit IEEE 802 MAC Individual address.

Values: Any

Name: WEPKeyMapIndex

Description: This RID is used to enable WEP Key-Mapping for transmitting and selects which of the two WEP Key-Mapping keys will be used by the AP firmware.

Format: Integer

Values: One of the following:

Value	Description
0	Enable WEP Key-Mapping and use WEP KeyMapKey0
1	Enable WEP Key-Mapping and use WEP KeyMapKey1
0xFF	Disable WEP Key-Mapping for the specified station.

4.4.2.2.23 BroadcastKeyID

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FCB8**	1
2	BroadcastKeyID	1

Field descriptions

Name: BroadcastKeyID

Description: Specifies which entry in the Default Key table (see 4.4.2.2.5 cnfDefaultKey0-3) to use for transmitted Broadcast/Multicast packets.

Format: Integer

Values: 2 or 3; Default = 2. Any value other than 2 or 3 will generate a error condition and will not be used.

An AP with WEP Key-Mapping enabled uses the first two entries in the Default Key table for transmission of station frames. Therefore, all Broadcast/Multicast frames must be encrypted using WEP Default key 2 or 3.

4.4.2.2.24 EntSecFlag

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FCB9**	1
2	EntSecFlag	1

Field descriptions

Name: EntSecFlags

Description: Specifies whether certain types of Intra-BSS Security Filtering are enabled

Format: Bitmask, with bits 0 and 1 defined as follows:

Bit	Definition
0	Block unauthenticated Intra-BSS routing
1	Block all Intra-BSS routing

Values: 0 = Disabled, 1 = Enabled

This RID provides the ability to block communications within the BSS. This security feature is intended for network configurations in which IEEE 802.1X authentication is used in conjunction with a Radius Server.

- Setting Bit 0 prevents Stations from communicating through the Access Point with other Stations in the BSS before the Station has been Authenticated.
- Setting Bit 1 prevents any communications between stations in the BSS. With this bit set, stations in the BSS can only communicate with resources on a backbone network bridged through the AP.

4.4.2.2.25 ScanRequest (Station Only)

Word Offset	Field name	Size (words)
0	RecordLen: 3	1
1	RID: **FCE1**	1
2	ChannelList	1
3	TxRate	1

Field Descriptions

Name: ChannelList

Description: ScanRequest notifies the firmware that the host desires a scan. The channelList is a bit map, where bit 0 corresponds to Channel 1, bit 1 to Channel 2 and so on through bit 13, corresponding to Channel 14. However, only channels allowed by the regulatory domain are actually scanned, regardless of bit settings.

Format: Bit-mapped integer

Values: any (reserved bits 0)

Name: TxRate

Description: This value determines the rate at which the Probe request frames will be sent, as follows:

Bit	Data Rate
0	1 Mbps
1	2 Mbps
2	5.5 Mbps
3	11 Mbps

The highest data rate specified will be used.

Format: Bit-mapped integer

Values: any (reserved bits 0)

4.4.2.2.26 JoinRequest (Station Only)

Word Offset	Field name	Size (words)
0	RecordLen: 5	1
1	RID: **FCE2**	1
2	BSSID	3
5	Channel	1

Field Descriptions

Name: BSSID

Description: Specifies the BSS to join.

Format: Byte Area

Values: Universal MAC Address.

Name: Channel

Description: Specifies the channel number of the BSS.

Format: Integer

Values: Range 0–14

The host must wait for the LinkStatus notification from the JoinRequest before issuing the next JoinRequest.

4.4.2.2.27 authenticateStation (AP Only)

Word Offset	Field name	Size (words)
0	RecordLen:6	1
1	RID: **FCE3**	1
2	Address	3
5	Status	1
6	Authentication Algorithm	1

Used when the host receives an AuthenticationRequest notification message and must respond with an authenticateStation. This allows the host to perform a roster security check on an authenticating station.

Field Descriptions

Name: Address

Description: This is the address of the station that is being authenticated.

Format: 48-bit address

Values: Any

Name: Status

Description: The status to return to the authenticating station. Any value other then SUCCESS (0) will result in denial of authentication. The status codes are defined in Reference [1].

Format: Integer

Values: Any

Name: Authentication Algorithm

Description: The authentication algorithm to use for authenticating the station. This value must be the same as the value retrieved in the AuthenticationRequest information frame (see 4.4.5.4.3 AuthenticationRequest (AP Only)).

Format: Integer

Values: One of the following:

0 – 802.11 Open System Authentication
1 – 802.11 Shared Key Authentication

4.4.2.2.28 ChannelInfoRequest (AP Only)

Word Offset	Field name	Size (words)
0	RecordLen: 3	1
1	RID: **FCE4**	1
2	ChannelList	1
3	ChannelDwellTime	1

ChannelInfoRequest notifies the firmware that the host desires a scan of the channels to determine the activity level in each channel. The result of this request is the channelInfoResult notification message. The firmware will not send any packets during this scanning process. The firmware will listen to each channel to collect the average and peak noise level, and to discover any operating 802.11 WLANs.

Field Descriptions

Name: ChannelList

Description: The channelList is a bit map, where bit 0 corresponds to Channel 1, bit 1 to Channel 2 and so on through bit 13, which corresponds to Channel 14. Only channels allowed by the regulatory domain will actually be scanned, irrespective of bit settings.

Format: Bit-mapped integer

Values: any (reserved bits 0)

Name: ChannelDwellTime

Description: The channelDwellTime is an integer value that controls the time, in msec, that the scanning process dwells on each channel.

Format: Integer

Values: Any

4.4.2.2.29 HostScan

Word Offset	Field Name	Size (words)
0	RecordLen: 4	1
1	RID: **FCE5**	1
2	channelList	1
3	txRate	1
4	targetSSID	1

HostScan notifies the firmware that the Host desires a non-destructive sweep of the available BSSIDs matching the specified target SSID. This sweep is “non-destructive” in that the firmware will maintain its current association with the BSS while it performs the sweep. When the sweep is completed, the Host Scan Table is returned to the Host in a HostScanResults information packet (see 4.4.5.2.4 HostScanResults (Station Only)). The last Host Scan Table can also be retrieved through the HostScanResults RID (see 4.4.3.2.20 HostScanResults).

Field Descriptions

Name: ChannelList

Description: The channelList is a bit map, where bit 0 corresponds to Channel 1, bit 1 to Channel 2 and so on through bit 13, which corresponds to Channel 14. Only channels allowed by the regulatory domain will actually be scanned, irrespective of bit settings.

Format: Bit-mapped integer

Values: any (reserved bits 0)

Name: txRate

Description: This value determines the rate at which Probe request frames are sent, as follows:

Bit	Data Rate
0	1 Mbps
1	2 Mbps
2	5.5 Mbps
3	11 Mbps

The highest data rate specified will be used.

Format: Bit-mapped integer

Values: any (reserved bits 0)

Name: targetSSID

Description: Identifies the Service Set ID for which the station is to sweep. A zero-length string indicates any SSID.

Format: Byte string

Values: Range (StringLen: 0–32)

4.4.2.3 Behavior Parameters

Behavior parameters contain values that may be changed to adjust the PRISM MAC behavior to the needs of the controlling program. Table 4.15 lists the behavior parameters.

Table 4.15: Behavior Parameters

RID	Name	Description
FCE0	TickTime	[PRI] Auxiliary Timer tick interval

4.4.2.3.1 TickTime

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FCE0**	1
2	TickTime	1

Field Descriptions

Name: TickTime

Description: Specifies the time in Kμs between Auxiliary timer ticks, as follows:

0 = Disable Tick event generation.
1–65535 = Number of Kμs between Tick events

Each timer tick generates a Tick event (see 4.1.2.4.1 EvStat Register (I/O offset 30)). The Auxiliary timer has no further effect on PRISM MAC behavior. This timer can be used as a timing source for Host software functions. If Tick event generation is already enabled (that is, the current value is not 0), a newly-written value becomes effective after the next Tick event occurs.

Format: Integer

Values: Default: 10

4.4.3 INFORMATION RECORDS

The contents of Information records can only be read: an attempt to write the Information records contents is not allowed.

After power-up/reset, Information record field values are always within the specified ranges, since at power-up/reset and during Initialize command execution (see 4.2.1.1 Initialize Command), the Information records are filled with their Fixed or Initial values. Fixed values remain unchanged, while Initial values may be changed by dynamic PRISM MAC behavior.

For some fields, a Range as well as a Fixed value is specified. This means the described component version for that field always contains the specific value, but that other versions of the same component may have any other value within the specified range.

- Information records are organized into three groups:
- NIC Information records
- MAC Information records
- RF modem Information records

The following sections describe the Information records.

4.4.3.1 NIC Information Records

NIC Information records reflect specific characteristics of the Network Interface Card (NIC), which is the hardware assembly in which the PRISM MAC is integrated. Table 4.16 lists the NIC Information records.

Table 4.16: NIC Information Records

RID	Name	Description
FD00	MaxLoadTime	[PRI] Maximum response time of the Download command.
FD01	DownloadBuffer	[PRI] Download buffer location and size.
FD02	PRIIdentity	[PRI] Primary Functions firmware identification.
FD03	PRISupRange	[PRI] Primary Functions interface; Supplier compatibility range.
FD04	CFIActRanges	[PRI] Controller interface; Actor compatibility ranges.
FD0A	NICSerialNumber	[PRI] Network Interface Card serial number.
FD0B	NICIdentity	[PRI] Network Interface Card identification.
FD0C	MFISupRange	[PRI] Modem interface, Supplier compatibility range.

Table 4.16: NIC Information Records (Continued)

FD0D	CFISupRange	[PRI] Controller interface; Supplier compatibility range.
FD10	ChannelList	Allowed communications channels.
FD11	RegulatoryDomains	List of intended regulatory domains.
FD12	TempType	Hardware temperature range code.
FD13	CIS	PC Card Standard Card Information Structure
FD20	STAIdentity	[STA] Station Functions firmware identification.
FD21	STASupRange	[STA] Station Functions interface; Supplier compatibility range.
FD22	MFIActRanges	[STA] Modem interface; Actor compatibility ranges.
FD23	CFIActRanges	[STA] Controller interface; Actor compatibility ranges.
FFFE	BuildSequence	Uniquely identifies created extended version number
FFFF	FWID	Host assignable identifier for the firmware, typically a file name

Several NIC Information records contain Component Identification and Compatibility Range values. Chapter 7 Production Data Area describes assignment guidelines and usage of these values.

Notice that actual Minor Version numbers are not specified. These fixed values are to be maintained by the firmware implementers in accordance with the guidelines given in 7.2 Identity and Compatibility Information. Changing such a value does not require a change of this definition.

4.4.3.1.1 MaxLoadTime

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FD00**	1
2	MaxLoadTime	1

Field Descriptions

Name: MaxLoadTime

Description: Specifies the maximum response time of the Download command in number of milliseconds. A properly functioning PRISM MAC completes the synchronous download process within this time, regardless of the ProgMode command control value specified (see 4.2.4.2 Download Command).

Format: Integer

Values: Range: 5–65535; Fixed: 5000

4.4.3.1.2 DownloadBuffer

Word Offset	Field name	Size (words)
0	RecordLen: 4	1
1	RID: **FD01**	1
2	DownloadBufferPage	1
3	DownloadBufferOffset	1
4	DownloadBufferLength	1

Field Descriptions

Name: DownloadBufferPage

Description: Specifies the Page address of the download buffer used for non-volatile memory programming by means of the Download command (see 4.2.4.2 Download Command). This Page address value can be written directly into the AuxPage register to set the Auxiliary Port for download buffer access.

Format: Integer

Values: Range: 0000–3FFF hexadecimal; Fixed: 0060 hexadecimal

Name: DownloadBufferOffset

Description: Specifies the Data offset of the download buffer. This Data offset value can be written directly into the AuxOffset register to set the Auxiliary Port for download buffer access.

Format: Integer

Values: Range: 0000–007E hexadecimal (even numbers only); Fixed: 0000 hexadecimal

Name: DownloadBufferLength

Description: Specifies the length of the download buffer in number of bytes.

Format: Integer

Values: Range: 0800–8000 hexadecimal (in steps of 0400 hexadecimal); Fixed: 0200 hexadecimal

4.4.3.1.3 PRIIdentity

Word Offset	Field name	Size (words)
0	RecordLen: 5	1
1	RID: **FD02**	1
2	PRICompID	1
3	PRIVariant	1
4	PRIMajorVersion	1
5	PRIMinorVersion	1

Field Descriptions

Name: PRICompID

Description: Specifies the Primary Functions firmware Component Identification number.

Format: Integer

Values: Range: 21–30; Fixed: 21

Name: PRIVariant

Description: Specifies the Variant number of the Primary Functions firmware component.

Format: Integer

Values: Range: 1–99; Fixed: 1

Name: PRIMajorVersion

Description: Specifies the Major Version of the Primary Functions firmware component.

Format: Integer

Values: Range: 1–99; Fixed: 1

Name: PRIMinorVersion

Description: Specifies the Minor Version number of the Primary Functions firmware component.

Format: Integer

Values: Range: 0–99

4.4.3.1.4 PRISupRange

Word Offset	Field name	Size (words)
0	RecordLen: 7	1
1	RID: **FD03**	1
2	PRIRole	1
3	PRIID	1
4	PRIVariant	1
5	PRIBottom	1
6	PRITop	1

Field Descriptions

Name: PRIRole

Description: Specifies the Role of the implemented Primary Functions firmware interface.

Format: Boolean

Values: Fixed: Off = Supplier

Name: PRIID

Description: Specifies the Identification number of the Primary Functions firmware interface.

Format: Integer

Values: 3

Name: PRIVariant

Description: Specifies the Variant number of the Primary Functions firmware interface.

Format: Integer

Values: Range: 1–99; Fixed: 1

Name: PRIBottom

Description: Specifies the lower Compatibility Range number of the Primary Functions firmware interface.

Format: Integer

Values: Range: 1–255; Fixed: 1

Name: PRITop

Description: Specifies the higher Compatibility Range number of the Primary Functions firmware interface.

Format: Integer

Values: Range: 1–255; Fixed: 1

4.4.3.1.5 CFIActRanges

Word Offset	Field name	Size (words)
0	RecordLen: 3+*n**3; *n*=1–20; Fixed: 6 (i.e., *n*=1)	1
1	RID: **FD04**	1
2	CFIRole	1
3	CFIID	1
4	CFIVariant	1
5	CFIBottom	1
6	CFITop	1

This record provides information on all interfaces between the Controller and the Primary Firmware. As many as 20 such interfaces may be implemented.

Refer to 7.2 Identity and Compatibility Information for a discussion of compatibility ranges.

Field Descriptions

Name: CFIRole

Description: Specifies the Role of the implemented Controller-Firmware interface.

Format: Boolean

Values: Fixed: On = Actor

Name: CFIID

Description: Specifies the Identification number of the Controller-Firmware interface.

Format: Integer

Values: 2

Name: CFIVariant

Description: Specifies the Variant number of the Controller-Firmware interface.

Format: Integer

Values: Range: 1–99; Fixed: 1

Name: CFIBottom

Description: Specifies the lower Compatibility Range number of the Controller-Firmware interface.

Format: Integer

Values: Range: 1–255; Fixed: 1

Name: CFITop

Description: Specifies the higher Compatibility Range number of the Controller-Firmware interface.

Format: Integer

Values: Range: 1–255; Fixed: 1

4.4.3.1.6 NICSerialNumber

Word Offset	Field name	Size (words)
0	RecordLen: 7	1
1	RID: **FD0A**	1
2	NICSerialNumber	6

Field Descriptions

Name: NICSerialNumber

Description: Specifies the unique serial number of the NIC.

Format: Byte Area

Values: Fixed: PDR 0003 - NIC Serial Number

4.4.3.1.7 NICIdentity

Word Offset	Field name	Size (words)
0	RecordLen: 5	1
1	RID: **FD0B**	1
2	NICCompID	1
3	NICVariant	1
4	NICMajorVersion	1
5	NICMinorVersion	1

Field Descriptions

Name: NICCompID

Description: Specifies the NIC Component Identification number.

Format: Integer

Values: Range: 1–20; Fixed: PDR 0008 - NIC Component Identification (1)

Name: NICVariant

Description: Specifies the Variant number of the NIC component.

Format: Integer

Values: Range: 1–99; Fixed: PDR 0008 - NIC Component Identification (1)

Name: NICMajorVersion

Description: Specifies the Major Version number of the NIC component.

Format: Integer

Values: Range: 1–99; Fixed: PDR 0008 - NIC Component Identification (1)

Name: NICMinorVersion

Description: Specifies the Minor Version number of the NIC component.

Format: Integer

Values: Range: 0–99; Fixed: PDR 0008 - NIC Component Identification (0)

4.4.3.1.8 MFISupRange

Word Offset	Field name	Size (words)
0	RecordLen: 6	1
1	RID: **FD0C**	1
2	MFIRole	1
3	MFIID	1
4	MFIVariant	1
5	MFIBottom	1
6	MFITop	1

Field Descriptions

Name: MFIRole

Description: Specifies the Role of the implemented RF modem-Firmware interface.

Format: Boolean

Values: Fixed: PDR 0006 - RF Modem Interface Compatibility Levels (Off)

Name: MFIID

Description: Specifies the Identification number of the RF modem-Firmware interface.

Format: Integer

Values: Fixed: PDR 0006 - RF Modem Interface Compatibility Levels (1)

Name: MFIVariant

Description: Specifies the Variant number of the RF modem-Firmware interface.

Format: Integer

Values: Range: 1–99; Fixed: PDR 0006 - RF Modem Interface Compatibility Levels (1)

Name: MFIBottom

Description: Specifies the lower Compatibility Range number of the RF modem-Firmware interface.

Format: Integer

Values: Range: 1–255; Fixed: PDR 0006 - RF Modem Interface Compatibility Levels (1)

Name: MFITop

Description: Specifies the higher Compatibility Range number of the RF modem-Firmware interface.

Format: Integer

Values: Range: 1–255; Fixed: PDR 0006 - RF Modem Interface Compatibility Levels (1)

4.4.3.1.9 CFISupRange

Word Offset	Field name	Size (words)
0	RecordLen: 6	1
1	RID: **FD0D**	1
2	CFIRole	1
3	CFIID	1
4	CFIVariant	1
5	CFIBottom	1
6	CFITop	1

This record provides information on the supplier compatibility range provided by the Controller to the Primary and Secondary Firmware.

Refer to 7.2 Identity and Compatibility Information for a discussion of compatibility ranges.

Field Descriptions

Name: CFIRole

Description: Specifies the Role of the implemented Controller-Firmware interface.

Format: Boolean

Values: Fixed: PDR 0007 - PRISM MAC Interface Compatibility Levels (Off)

Name: CFIID

Description: Specifies the Identification number of the Controller-Firmware interface.

Format: Integer

Values: Fixed: PDR 0007 - PRISM MAC Interface Compatibility Levels (2)

Name: CFIVariant

Description: Specifies the Variant number of the Controller-Firmware interface.

Format: Integer

Values: Range: 1–99; Fixed: PDR 0007 - PRISM MAC Interface Compatibility Levels (1)

Name: CFIBottom

Description: Specifies the lower Compatibility Range number of the Controller-Firmware interface.

Format: Integer

Values: Range: 1–255; Fixed: PDR 0007 - PRISM MAC Interface Compatibility Levels (1)

Name: CFITop

Description: Specifies the higher Compatibility Range number of the Controller-Firmware interface.

Format: Integer

Values: Range: 1–255; Fixed: PDR 0007 - PRISM MAC Interface Compatibility Levels (1)

4.4.3.1.10 ChannelList

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FD10**	1
2	ChannelList	1

Field Descriptions

Name: ChannelList

Description: Specifies the communications channels that can be used with this NIC.

Format: Bit Area:

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	0	0	Chan 14	Chan 13	Chan 12	Chan 11	Chan 10	Chan 9	Chan 8	Chan 7	Chan 6	Chan 5	Chan 4	Chan 3	Chan 2	Chan 1

Each Chan*nn* bit corresponds to a specific communications frequency channel, where *nn* indicates the CHNL_ID number (Reference [1], section 15.4.6.2.):

On = allowed
Off = not allowed

Values: Fixed: PDR 0104 - Allowed Channel Bitmap (1FFF hexadecimal)

4.4.3.1.11 RegulatoryDomains

Word Offset	Field name	Size (words)
0	RecordLen: 7	1
1	RID: **FD11**	1
2	RegulatoryDomains	6

Field Descriptions

Name: RegulatoryDomains

Description: Specifies the regulatory domains for which the NIC is intended. See Reference [1], section 13.1.4.2., for the domain identification values.

Format: Byte String

Values: Range: StringLen: 1–10; Fixed: PDR 0103 - Regulatory Domain List (StringLen = 2, Byte Area = 30, 31 hex)

4.4.3.1.12 TempType

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FD12**	1
2	TempType	1

Field Descriptions

Name: TempType

Description: Specifies the operational temperature range of the NIC:

0001 hexadecimal = Type 1 = Commercial range 0 ⇔ +40 °C
0002 hexadecimal = Type 2 = Industrial range -20 ⇔ +55 °C
0003 hexadecimal = Type 3 = Industrial range -30 ⇔ +70 °C

Format: Integer

Values: Range: 0001–0003 hex; Fixed: PDR 0107 - Temperature Type (0001 hexadecimal)

4.4.3.1.13 CIS

Word Offset	Field name	Size (words)
0	RecordLen: 241	1
1	RID: **FD13**	1
2	CIS	240

Field Descriptions

Name: CIS

Description: Contains the PC Card Standard Card Information Structure (CIS) of the NIC, which describes the function and characteristics of the NIC. This structure is a linked list of data blocks, or tuples. The list starts at the first byte, and specifies its own length by means of an END tuple. See Reference [4] for the structure definition and the meaning of specific tuple values.

Format: Byte Area

Values: Fixed: Obtained from the Compact CIS Area, which is initially loaded with the Default CIS, and later re-loaded with a specific CIS deliverable.

4.4.3.1.14 STAIdentity

Word Offset	Field name	Size (words)
0	RecordLen: 5	1
1	RID: **FD20**	1
2	STACompID	1
3	STAVariant	1
4	STAMajorVersion	1
5	STAMinorVersion	1

Field Descriptions

Name: STACompID

Description: Specifies the Station Functions and AP Functions firmware Component Identification number. (Since each AP is also a Station, the values are overloaded into the same field in the case where AP functions are also present.)

Format: Integer

Values: 31 for Station functions; 331 for AP functions.

Name: STAVariant

Description: Specifies the Variant number of the Station Functions firmware component.

Format: Integer

Values: Range: 1–99; Fixed: 1

Name: STAMajorVersion

Description: Specifies the Major Version number of the Station Functions firmware component.

Format: Integer

Values: Range: 1–99; Fixed: 1

Name: STAMinorVersion

Description: Specifies the Minor Version number of the Station Functions firmware component.

Format: Integer

Values: Range: 0–99

4.4.3.1.15 STASupRange

Word Offset	Field name	Size (words)
0	RecordLen: 6	1
1	RID: **FD21**	1
2	STARole	1
3	STAID	1
4	STAVariant	1
5	STABottom	1
6	STATop	1

Field Descriptions

Name: STARole

Description: Specifies the Role of the implemented Station Functions firmware interface.

Format: Boolean

Values: Fixed: Off = Supplier

Name: STAID

Description: Specifies the Identification number of the Station Functions and AP Functions firmware interface. (Since each AP is also a Station, the values are overloaded into the same field in the case where AP functions are also present.)

Format: Integer

Values: 4 for Station; 5 for AP.

Name: STAVariant

Description: Specifies the Variant number of the Station Functions firmware interface.

Format: Integer

Values: Range: 1–99; Fixed: 1

Name: STABottom

Description: Specifies the lower Compatibility Range number of the Station Functions firmware interface.

Format: Integer

Values: Range: 1–255; Fixed: 1

Name: STATop

Description: Specifies the higher Compatibility Range number of the Station Functions firmware interface.

Format: Integer

Values: Range: 1–255; Fixed: 1

4.4.3.1.16 MFIActRanges

Word Offset	Field name	Size (words)
0	RecordLen: 3+*n**3; *n*=1–20; Fixed: 6 (i.e., *n*=1)	1
1	RID: **FD22**	1
2	MFIRole	1
3	MFIID	1
4	MFIVariant	1
5	MFIBottom	1
6	MFITop	1

This record provides information on all implemented RF modem-Hardware interfaces. As many as 20 interfaces may be implemented.

Refer to 7.2 Identity and Compatibility Information for a discussion of compatibility ranges.

Field Descriptions

Name: MFIRole

Description: Specifies the Role of the implemented RF modem-Firmware interface.

Format: Boolean

Values: Fixed: On = Actor

Name: MFIID

Description: Specifies the Identification number of the RF modem-Firmware interface.

Format: Integer

Values: 1

Name: MFIVariant

Description: Specifies the Variant number of the RF modem-Firmware interface.

Format: Integer

Values: Range: 1–99; Fixed: 1

Name: MFIBottom

Description: Specifies the lower Compatibility Range number of the RF modem-Firmware interface.

Format: Integer

Values: Range: 1–255; Fixed: 1

Name: MFITop

Description: Specifies the higher Compatibility Range number of the RF modem-Firmware interface.

Format: Integer

Values: Range: 1–255; Fixed: 1

4.4.3.1.17 CFIActRanges

Word Offset	Field name	Size (words)
0	RecordLen: 3+*n**3; *n*=1–20; Fixed: 6 (i.e., *n*=1)	1
1	RID: **FD23**	1
2	CFIRole	1
3	CFIID	1
4	CFIVariant	1
5	CFIBottom	1
6	CFITop	1

This record provides information on all interfaces between the Controller and the Secondary Firmware. As many as 20 such interfaces may be implemented.

Refer to 7.2 Identity and Compatibility Information for a discussion of compatibility ranges.

Field Descriptions

Name: CFIRole

Description: Specifies the Role of the implemented Controller-Firmware interface.

Format: Boolean

Values: Fixed: On = Actor

Name: CFIID

Description: Specifies the Identification number of the Controller-Firmware interface.

Format: Integer

Values: 2

Name: CFIVariant

Description: Specifies the Variant number of the Controller-Firmware interface.

Format: Integer

Values: Range: 1–99; Fixed: 1

Name: CFIBottom

Description: Specifies the lower Compatibility Range number of the Controller-Firmware interface.

Format: Integer

Values: Range: 1–255; Fixed: 1

Name: CFITop

Description: Specifies the higher Compatibility Range number of the Controller-Firmware interface.

Format: Integer

Values: Range: 1–255; Fixed: 1

4.4.3.1.18 ProductName

Word Offset	Field name	Size (words)
0	RecordLen: 3	1
1	RID: **FD24**	1
2	ProductName	9

Field Descriptions

Name: ProductName

Description: Used to obtain the Part Number production data record (see 7.4.1 PDR 0x0001 - Manufacturing Part Number). This is a read-only value, and does not affect the behavior of the firmware.

Format: Byte string (length encoded)

Values: Range: StringLen 1—16 characters

4.4.3.1.19 BuildSequence

Word Offset	Field name	Size (words)
0	RecordLen: 3	1
1	RID: **FFFE**	1
2	Primary BuildSeq	1
3	Secondary/Tertiary BuildSeq	1

Field Descriptions

Name: BuildSequence

Description: Used to uniquely identify the version information. Used with version RID FD20 to create an extended version number.

NOTE: This field is set to zero for released firmware builds. A non-zero value indicates an in-house, non-released (developer's) build of firmware.

Format: Integer

Values: Any

4.4.3.1.20 Firmware Identifier (FWID)

Word Offset	Field name	Size (words)
0	RecordLen: 15	Up to 15
1	RID: **FFFF**	1
2	Primary Identifier	7
9	Secondary or Tertiary Identifier	7

Field Descriptions

Name: FWID

Description: This is a Host-assignable identifier for the firmware, and is typically a file name. Intersil flash programming tools use this area to plug the filename of the firmware image being downloaded in an 8.3 format in each field. This comes from plug information that is queued from the host and may contain variable information. See 8.1.3.1 S3 Plug Records, for additional information.

Format: Binary

Values: Variable

4.4.3.2 MAC Information Records

MAC Information records specify Media Access Control attributes. These are current parameter values as well as dynamic operation results intended for behavior management and diagnostic purposes.

Table 4.17 lists the MAC Information Records.

Table 4.17: MAC Information Records

RID	Name	Description
FD40	PortStatus	[STA] Actual MAC Port connection control status.
FD41	CurrentSSID	[STA] Identification of the actually connected SS.
FD42	CurrentBSSID	[STA] Identification of the actually connected BSS.
FD43	CommsQuality	[STA] Quality of the Basic Service Set connection.
FD44	CurrentTxRate	[STA] Actual transmit data rate.
FD45	CurrentBeaconInterval	Interval time for Beacon transmissions and receptions.
FD46	CurrentScaleThresholds	Actual System Scale thresholds settings.
FD47	ProtocolRspTime	Maximum time to await a response to a request message.
FD48	ShortRetryLimit	Maximum number of transmit attempts for short frames.
FD49	LongRetryLimit	Maximum number of transmit attempts for long frames.
FD4A	MaxTransmitLifetime	Maximum transmit frame handling duration.
FD4B	MaxReceiveLifetime	Maximum received frame handling duration.
FD4C	CFPollable	[STA] Contention Free pollable capability indication.
FD4D	AuthenticationAlgorithms	Available Authentication Algorithms indication.
FD4F	PrivacyOptionImplemented	WEP Option availability indication.
FD51	dBmCommsQuality	dBm normalized values for CommsQuality
FD80	CurrentTxRate1	[AP] Actual Port 1 transmit data rate.
FD81	CurrentTxRate2	[AP] Actual Port 2 transmit data rate.
FD82	CurrentTxRate3	[AP] Actual Port 3 transmit data rate.
FD83	CurrentTxRate4	[AP] Actual Port 4 transmit data rate.
FD84	CurrentTxRate5	[AP] Actual Port 5 transmit data rate.
FD85	CurrentTxRate6	[AP] Actual Port 6 transmit data rate.
FD86	OwnMACAddress	[AP] Unique local node MAC Address.
FD88	ScanResults	Passes last sweep results gathered to the host on demand through RID mechanism.
FD89	HostScanResults	Returns results of last Host Scan Request.
FD8A	AuthenicationUsed	Authentication algorithm used in current association.

4.4.3.2.1 PortStatus

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FD40**	1
2	PortStatus	1

Field Descriptions

Name: PortStatus

Description: Specifies the current MAC Port connection status, as follows:

1 = Disabled
2 = Searching for initial connection
3 = Connected to IBSS
4 = Connected to ESS (see note)
5 = Out of Range (in ESS)
6 = Connected to WDS
8 = Started Host AP

NOTE: Firmware still maintains a connection even if the connection status is out-of-range.

Format: Integer

Values: 1–7, 8; Initial: 1

4.4.3.2.2 CurrentSSID

Word Offset	Field name	Size (words)
0	RecordLen: 18	1
1	RID: **FD41**	1
2	CurrentSSID	17

Field Descriptions

Name: CurrentSSID

Description: Specifies the Service Set to which the station is currently connected. Notice that this information is only relevant if the MAC Port is in the Enabled state (see 4.2.1.2 Enable Command).

Format: Byte String

Values: Range: StringLen: 0–32; Initial: StringLen = 0

4.4.3.2.3 CurrentBSSID

Word Offset	Field name	Size (words)
0	RecordLen: 4	1
1	RID: **FD42**	1
2	CurrentBSSID	3

Field Descriptions

Name: CurrentBSSID

Description: Specifies the MAC Address used as the identifier of the Basic Service Set to which the station is currently connected. Notice that this information is only relevant if the MAC Port is in the Enabled state (see 4.2.1.2 Enable Command).

Format: Byte Area

Values: Initial: 00, 00, 00, 00, 00, 00 hexadecimal

4.4.3.2.4 CommsQuality

Word Offset	Field name	Size (words)
0	RecordLen: 4	1
1	RID: **FD43**	1
2	CQ.currBSS	1
3	ASL.currBSS	1
4	ANL.currFC	1

This record indicates the quality of communications within the current BSS. The firmware derives the values for the fields of this record based on signal level and noise level measurements. During normal operation these measurements are continuously updated by the firmware, which then calculates CQ.currBSS based on a signal-to-noise ratio (SNR). The firmware uses this value in conjunction with the cnfSystemScale Configuration Record (see 4.4.2.1.7 cnfSystemScale) to initiate a scan for an AP offering better communications quality.

Field Descriptions

Name: CQ.currBSS

Description: Specifies the Communications Quality of the Basic Service Set to which the station is currently connected. Notice that this information is only relevant if the MAC Port is in the Enabled state (see 4.2.1.2 Enable Command).

Format: Integer

Values: Range: 0–92; Initial: 0

Name: ASL.currBSS

Description: Specifies the Average Signal Level of the Basic Service Set to which the station is currently connected. Notice that this information is only relevant if the MAC Port is in the Enabled state (see 4.2.1.2 Enable Command).

Format: Integer

Values: Range: 27–154; Initial: 27

Name: ANL.currFC

Description: Specifies the Average Noise Level of the Frequency Channel currently used for communications. Notice that this information is only relevant if the MAC Port is in the Enabled state (see 4.2.1.2 Enable Command).

Format: Integer

Values: Range: 27–154; Initial: 27

4.4.3.2.5 CurrentTxRate

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FD44**	1
2	CurrentTxRate	1

Field Descriptions

Name: CurrentTxRate

Description: Specifies the data rate in Mbit/s that is currently used for message transmission. Notice that this information is only relevant if the MAC Port is in the Enabled state (see 4.2.1.2 Enable Command).

Format: Integer

Values: Range: 1–100; Initial: 2

4.4.3.2.6 CurrentBeaconInterval

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FD45**	1
2	CurrentBeaconInterval	1

Field Descriptions

Name: CurrentBeaconInterval

Description: Specifies the interval time in Kμs used for scheduling Beacon transmissions and/or expecting Beacon receptions.

Format: Integer

Values: Range: 1–65535; Initial: 100

4.4.3.2.7 CurrentScaleThresholds

Word Offset	Field name	Size (words)
0	RecordLen: [STA] 6; [AP] 4	1
1	RID: **FD46**	1
2	EnergyDetectThreshold	1
3	CarrierDetectThreshold	1
4	DeferThreshold	1
5	[STA] CellSearchThreshold	1
6	[STA] OutOfRangeThreshold	1
7	[STA] DeltaSNR	1

Field Descriptions

Name: EnergyDetectThreshold

Description: Specifies the Energy Detect Threshold value used by the receiver hardware. Notice that this information is only relevant if the MAC Port (for an AP, Port 0) is in the Enabled state (see 4.2.1.2 Enable Command).

Format: Integer

Values: Range: 48–89; Initial: 54

Name: CarrierDetectThreshold

Description: Specifies the Carrier Detect Threshold value used by the receiver hardware. Notice that this information is only relevant if the MAC Port (for an AP, Port 0) is in the Enabled state (see 4.2.1.2 Enable Command).

Format: Integer

Values: Range: 48–89; Initial: 54

Name: DeferThreshold

Description: Specifies the Defer Threshold value used by the receiver hardware. Notice that this information is only relevant if the MAC Port is in the Enabled state (see 4.2.1.2 Enable Command).

Format: Integer

Values: Range: 48–89; Initial: 54

Name: CellSearchThreshold

Description: Specifies the Cell Search Threshold value used by the Roaming function of the receiver process to determine whether it is necessary to search for a better AP. Notice that this information is only relevant if the MAC Port is in the Enabled state (see 4.2.1.2 Enable Command).

Format: Integer

Values: Range: 0–154; Initial: 10

Name: OutOfRangeThreshold

Description: Specifies the Out of Range Threshold value used by the Roaming function of the receiver process to determine whether it is necessary to search for any other AP. Notice that this information is only relevant if the MAC Port (for an AP, Port 0) is in the Enabled state (see 4.2.1.2 Enable Command).

Format: Integer

Values: Range: 0–154; Initial: 2

Name: DeltaSNR

Description: Threshold value used by a station during roaming. While roaming, a station will not associate with a new AP unless that AP offers an increase in CommsQuality (see 4.4.3.2.4 CommsQuality) equal to or greater than this value.

Format: Integer

Values: Any

4.4.3.2.8 ProtocolRspTime

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FD47**	1
2	ProtocolRspTime	1

Field Descriptions

Name: ProtocolRspTime

Description: Specifies the Protocol Response Time in Kμs used by the transmitter process. This is the maximum time the transmitter process waits for a response to a management request and to PS-Poll.

Format: Integer

Values: Range: 1–1000; Fixed: 100

4.4.3.2.9 ShortRetryLimit

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FD48**	1
2	ShortRetryLimit	1

Field Descriptions

Name: ShortRetryLimit

Description: Specifies the Retry Limit (that is, the maximum number of transmit attempts) applied by the transmitter process for transmission of a frame whose length is less than or equal to RTSThreshold.

Format: Integer

Values: Range: 1–15; Fixed: 7

4.4.3.2.10 LongRetryLimit

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FD49**	1
2	LongRetryLimit	1

Field Descriptions

Name: LongRetryLimit

Description: Specifies the Retry Limit (that is, the maximum number of transmit attempts) applied by the transmitter process for transmission of a frame whose length is greater than RTSThreshold.

Format: Integer

Values: Range: 1–15; Fixed: 4

4.4.3.2.11 MaxTransmitLifetime

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FD4A**	1
2	MaxTransmitLifetime	1

Field Descriptions

Name: MaxTransmitLifetime

Description: Specifies the Maximum Transmit Lifetime in Kμs used by the transmitter process, during which attempts are made to transmit a particular Transmit Frame provided by the Host software. If the time expires before the frame is successfully transmitted, that frame is dropped.

Format: Integer

Values: Range: 1–1000; Fixed: 512

4.4.3.2.12 MaxReceiveLifetime

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FD4B**	1
2	MaxReceiveLifetime	1

Field Descriptions

Name: MaxReceiveLifetime

Description: Specifies the Maximum Receive Lifetime in Kμs used by the receiver process during which attempts are made to reassemble a particular message after the first fragment is received. If the time expires before the frame is completely received, the frame reception is terminated and the related information is dropped.

Format: Integer

Values: Range: 1–1000; Fixed: 512

4.4.3.2.13 CFPollable

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FD4C**	1
2	CFPollable	1

Field Descriptions

Name: CFPollable

Description: Specifies whether the Station is able to respond with a data frame to a CF-POLL frame within a SIFS time. If False, the Station is not always able to respond within that time.

Format: Boolean

Values: Fixed: No

4.4.3.2.14 AuthenticationAlgorithms

Word Offset	Field name	Size (words)
0	RecordLen: *n**2+1; *n* = 1–6; Fixed: 3 (i.e., *n* = 1)	1
1	RID: **FD4D**	1
2	AuthenticationType	1
3	TypeEnabled	1

This record provides information about supported authentication algorithms. As many as 6 authentication algorithms may be implemented.

Field Descriptions

Name: AuthenticationType

Description: Specifies the authentication algorithm supported by this node, as follows:

1 = Open System
2 = Shared Key

Format: Integer

Values: Range: 1–2; Fixed: 1

Name: TypeEnabled

Description: Specifies whether support for the preceding AuthenticationType is actually enabled.

Format: Boolean

Values: Fixed: Yes

4.4.3.2.15 PrivacyOptionImplemented

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FD4F**	1
2	PrivacyOptionImplemented	1

Field Descriptions

Name: PrivacyOptionImplemented

Description: Specifies whether the Privacy Option is available to generate and handle messages encrypted according to the Wired Equivalent Privacy (WEP) conventions. If False, no Privacy Option is available.

Format: Boolean

Values: Fixed: Yes

4.4.3.2.16 dBmCommsQuality

Word Offset	Field name	Size (words)
0	RecordLen: 4	1
1	RID: **FD51**	1
2	CQdb.currBSS	1
3	ASLdb.currBSS	1
4	ANLdb.currFC	1

This record provides a dBm-normalized value of the quality of communications within the current BSS. It is similar to CommsQuality, RID 43 (see 4.4.3.2.4 CommsQuality). However, note that dBm-normalized values are negative numbers.

Field Descriptions

Name: CQdb.currBSS

Description: Reflects the Communications Quality of the Basic Service Set to which the station is currently connected.

Format: Signed Integer

Values: Range, based on range of the radio

Name: ASLdb.currBSS

Description: Reflects the Average Signal Level of the Basic Service Set to which the station is currently connected.

Format: Signed Integer

Values: Range, based on the capabilities of the radio

Name: ANLdb.currFC

Description: Reflects the Average Noise Level of the Frequency Channel currently used for communications.

Format: Signed Integer

Values: Range, based on range of the radio

4.4.3.2.17 CurrentTxRate1–6

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FD80–FD85**	1
2	CurrentTxRate*N*	1

Field Descriptions

Name: CurrentTxRate*N*

Description: Specifies the data rate in Mbit/s currently used for message transmission. Notice that the lowest RID code indicates the lowest *N* value (1), and each next higher RID code indicates a next higher *N* value. Notice that this information is only relevant if MAC Port *N* is in the Enabled state (see 4.2.1.2 Enable Command).

Format: Integer

Values: Range: 1–100; Initial: 2

4.4.3.2.18 OwnMACAddress

Word Offset	Field name	Size (words)
0	RecordLen: 4	1
1	RID: **FD86**	1
2	OwnMACAddress	3

Field Descriptions

Name: OwnMACAddress

Description: Specifies the MAC Address that identifies this access point for data transfer.

Format: Byte Area

Values: Default: PDR 0101- Universal MAC Address (00, 00, 00, 00, 00, 00 hexadecimal)

4.4.3.2.19 ScanResults

Word Offset	Field name	Size (words)
0	FrameLen: 3+*n**31, where *n*=0–32	1
1	RID: FD88	1
2	Reserved for future use	1
3	Scan Reason	1
4	Scan Result 1 (optional)	31
	...	
35 + (*n*-1) x 31	Scan Result *n* (optional)	31

Field Descriptions

Name: Scan Reason

Description: Specifies the reason this scan was initiated, where:

0 = Scan Results Table not available due to scan in progress
1 = Host initiated
2 = Firmware initiated due to communications quality level
3 = Inquiry request from host for current SRT

Name: Scan Result *n*

Description: Each Scan Result field (when present) gives the scan result information for one Basic Service Set (BSS) (Station Only).

The actual number of Scan Result fields depends on the number of BSSs detected during the Scan. The Scan Result fields are sorted as follows:

- Extended Service Set (ESS) Scan Results come first, in descending order by signal-to-noise ratio
- Independent Basic Service Set (IBSS) results come after the ESS results, also in descending order of signal-to-noise ratio

The scan results are based on Probe Response frames received.

The format of each Scan Result (Station only) sub-field is:

Word Offset	Field name	Size (words)
0	SRT.CHID	1
1	SRT.ANL	1
2	SRT.SL	1
3	SRT.BSSID	3
6	SRT.BcnInt	1
7	SRT.Cap	1
8	SRT.SSID	17
25	SRT.SupRates	5
30	SRT.Rate	1

Sub-field Descriptions

Name: SRT.CHID

Description: BSS Channel ID from Probe Response

Format: Integer

Values: Range: 1–14

Name: SRT.ANL

Description: Average Noise Level of the Probe Responses received during the scan, averaged on a per-channel basis.

Format: Integer

Values: Range: 001B–009A hexadecimal

Name: SRT.SL

Description: Signal Level on which the Probe Response was received.

Format: Integer

Values: Range: 001B–009A hexadecimal

Name: SRT.BSSID

Description: MAC address of BSS responder from Probe Response.

Format: MAC address

Values: Any

Name: SRT.BcnInt

Description: BSS Beacon Interval (in Kμs) from Probe Response.

Format: Integer

Values: Any

Name: SRT.Cap

Description: BSS Capability information, copied from the Probe Response.

Format: Bit-mapped integer

Values: See Section 7.3.1.4 of Reference [1].

Name: SRT.SSID

Description: Copy of the Service Set Identifier element from the Probe Response. That is, the first word contains the length (the element type is 0) in the high-order octet, followed by 0 to 32 octets that contain the SSID string itself.

Format: String

Values: Length word, followed by string of length 0–32

Name: SRT.SupRates

Description: Copy of the contents Supported Rates element from the Probe Response. This sub-field is terminated with a zero stamp word. Each data rate is encoded as defined in Section 7.3.2.2 of Reference [1].

Format: Byte string

Values: Any

Name: SRT.Rate

Description: Data rate of the received Probe Response. The data rate is encoded as defined in Section 18.2.3.3 of Reference [7], where:

0x0A = 1 Mb/s
0x14 = 2 Mb/s
0x37 = 5.5 Mb/s
0x6E = 11 Mb/s

Format: Byte string

Values: Any

4.4.3.2.20 HostScanResults

HostScanResults returns the current Host Scan Results table maintained by the firmware. This table represents the results from the last host scan request initiated by RID FCE5 (see 4.4.2.2.29 HostScan).

Word Offset	Field name	Size (words)
0	FrameLen: 3+*n**32, where *n*=0–32	1
1	RID: **FD89**	1
2	resultSize	1
3	resultAvailable	1
4	Host Scan Result 1 (optional)	32
	...	
35 + (*n*-1) x 32	Host Scan Result *n* (optional)	32

Field Descriptions

Name: resultSize

Description: The size of each Host Scan Result entry. Host software uses this field as an offset into the table for the next entry.

Format: Integer

Values: 31

Name: resultAvailable

Description: Indicates whether the query for the Host Scan Results table was successful.

Format: Integer

Values: The following values may be returned:

0 = Host Scan Results table not available due to scan in progress
3 = Host Scan Results table is valid

Name: Host Scan Result *n*

Description: Each Scan Result field (when present) gives the scan result information for one Basic Service Set (BSS) (Station Only). Refer to 4.4.5.2.4 HostScanResults (Station Only) for details on the contents of these fields.

4.4.3.2.21 AuthenticationUsed

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: FD8A	1
2	authenticationAlgorithm	1

Field Description

Name: authenticationAlgorithm

Description: Identifies the authentication algorithm used in the current association. The algorithm numbers are defined in Reference [1].

Format: Integer

Values: One of the following:

0 = Open System Authentication
1 = Shared Key Authentication
0xFFFF = none

4.4.3.3 Modem Information Records

Modem Information reflects current parameter values as well as dynamic operation results intended for behavior management and diagnostic purposes.

Table 4.18 lists the Modem Information records.

Table 4.18: Modem Information Records

RID	Name	Description
FDC0	PHYType	Physical layer type indication.
FDC1	CurrentChannel	Actual frequency channel used for transmission.
FDC2	CurrentPowerState	Actual power consumption status.
FDC3	CCAMode	Clear channel assessment mode indication.
FDC6	SupportedDataRates	Data rates capability information.

For a more detailed description of the meaning and derivation of the information items, see the Configuration Entities and Information Entities sections of Reference [2] and Reference [3].

4.4.3.3.1 PHYType

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FDC0**	1
2	PHYType	1

Field Descriptions

Name: PHYType

Description: Indicates the Physical Layer Specification used for data transfer:

1 = fhss2dot4GHz
2 = dsss2dot4GHz
3 = irbaseband

Format: Integer

Values: Range: 1–3; Fixed: 2

4.4.3.3.2 CurrentChannel

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FDC1**	1
2	CurrentChannel	1

Field Descriptions

Name: CurrentChannel

Description: Specifies the CHNL_ID number of the communications channel currently used by the Basic Service Set to which the node is connected. This is always the number of an allowed communications channel (see 4.4.3.1.10 ChannelList). Notice that this information is only relevant if at least one MAC Port is in the Enabled state (see 4.2.1.2 Enable Command).

Format: Integer

Values: Range: 1–14; Initial: PDR 0105 - Default channel number (3)

4.4.3.3.3 CurrentPowerState

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FDC2**	1
2	CurrentPowerState	1

Field Descriptions

Name: CurrentPowerState

Description: Specifies the current power consumption state of the RF-Modem:

1 = Active
2 = Doze

Format: Integer

Values: Range: 1–2; Initial 2

4.4.3.3.4 CCAMode

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	RID: **FDC3**	1
2	CCAMode	1

Field Descriptions

Name: CCAMode

Description: Specifies the Clear Channel Assessment mode implemented by this node:

1 = Energy detect only
2 = Carrier sense only
3 = Energy detect and carrier sense

Format: Integer

Values: Range: 1–3; Fixed: 2

4.4.3.3.5 SupportedDataRates

Word Offset	Field name	Size (words)
0	RecordLen: 6	1
1	RID: **FDC6**	1
2	SupportedDataRates	5

Field Descriptions

Name: SupportedDataRates

Description: Specifies the data transfer rates supported by the RF-Modem, in units of 500 kbits/s.

Format: Byte String

Values: Range: StringLen: 1–10; Fixed: StringLen = 2 and Bytes = 2, 4

4.4.4 RECORD CONTENTS INITIAL VALUES

All Record fields will have a defined value after power-up/reset and Initialize Command execution. To support implementation, the origin of each of these values, and if possible, the actual values and their meaning are summarized in the following tables. In these tables, the following Initial Origin types are listed.

Compact CIS Area: This is a specific non-volatile memory area containing a PC Card Standard Card Information Structure data pattern (see Reference [4]). Initially this area is loaded with PRI Source constants (i.e., the Default CIS), and later it is re-loaded with a specific CIS deliverable.

Fixed Configuration: This is a special type of Source constant. Fixed Configurations have all Source constant characteristics, but they are named differently by Reference [2] and Reference [3] to indicate that their value plays a configuration role for the dynamic PRISM MAC behavior. These values are subject to (experimental) change during system testing, and, therefore, an implementation that enables a temporary change with use of an engineering tool (that is, without component re-assembly) is preferred.

PDR nnnn: This indicates values that are obtained from PDR with number *nnnn*. These PDRs, as well as the 'plugging' process that puts the values into the firmware component, are part of the firmware installation process. The related utilities will currently plug each of the defined PDR values. Future utilities may not plug some values because of migration reasons. To prevent undefined values in this case, for each of these values a Source constant is defined for replacement by the plugging process.

Source constant: This constant is defined in a firmware source file. Therefore, to change such a value, the source file must be changed and assembled to generate the component again. For all Source constants a value is defined, except for the Minor Version constants: for these 'Impl. value' is defined. This means that this value shall be maintained by the firmware implementers in accordance with the guidelines given in 1.1 Introduction. Changing such a value does not require a change of this definition.

4.4.4.1 Configuration Records Initial Values

Table 4.19 lists the initial values for Static Configuration entities.

Table 4.19: Network Parameters, Static Configuration Entities – Initial Values

RID	Name	Initial Value
FC00	cnfPortType	STA Source constant: 3 = Pseudo IBSS
FC01	cnfOwnMACAddress	PDR 0101: Universal MAC Address STA Source constant: 00-00-00-00-00-00 hex
FC02	cnfDesiredSSID	STA Source constant: StringLen = 0

Table 4.19: Network Parameters, Static Configuration Entities – Initial Values (Continued)

RID	Name	Initial Value
FC03	cnfOwnChannel	PDR 0105: Default channel number STA and AP Source constant: 3
FC04	cnfOwnSSID	STA and AP Source constant: StringLen = 21 Byte Area = “non-specified SSID !!”
FC05	cnfOwnATIMWindow	STA Source constant: 0
FC06	cnfSystemScale	STA and AP Source constant: 1 = Low density
FC07	cnfMaxDataLength	STA Source constant: 2304 AP Source constant: 1500
FC08	cnfWDSAddress	STA Source constant: 00-00-00-00-00-00 hex
FC0B	cnfMulticastReceive	STA Source constant: 1 = Yes
FC0C	cnfMaxSleepDuration	STA Source constant: 1000 Kμs
FC0E	cnfOwnName	STA and AP Source constant: StringLen = 0
FC10	cnfOwnDTIMPeriod	AP Source constant: 1
FC11	cnfWDSAddress1	AP Source constant: 00-00-00-00-00-00 hex
FC12	cnfWDSAddress2	AP Source constant: 00-00-00-00-00-00 hex
FC13	cnfWDSAddress3	AP Source constant: 00-00-00-00-00-00 hex
FC14	cnfWDSAddress4	AP Source constant: 00-00-00-00-00-00 hex
FC15	cnfWDSAddress5	AP Source constant: 00-00-00-00-00-00 hex
FC16	cnfWDSAddress6	AP Source constant: 00-00-00-00-00-00 hex
FC17	cnfMulticastPMBuffering	AP Source constant: 1 = Yes
FC2B	cnfMaxAssociatedStations	AP Source Constant: 250
FC2D	cnfRoamingMode	STA: 1 = Roaming decision made by firmware
FC2E	cnfHostAuthentication	AP Source Constant: False
FC30	cnfRcvCrcError	STA/AP Source Constant: False
FC34	cnfAPPCFInfo	AP Source Constant: 100 cnfCFPPeriod: 1 cnfCFPMaxDuration: 0 cnfCFPFlags: 0 = poll stations
FC35	cnfSTAPCFInfo	STA Source Constant: 0 = not requesting to be on pollList
FC40	cnfTIMCtrl	STA/AP Source Constant: *nn*: 0–0x1f 0x80*nn*=Set the TIM bit for AID 0x0zz 0x00*nn*=Clear the TIM bit for AID 0x0zz
FC42	cnfThirty2Tally	STA/AP Source Constant: False (0)
FC43	cnfEnhSecurity	AP Source Constant: 0
FC46	cnfDbmAdjust	STA/AP Source Constant: 100
FCB0	cnfShortPreamble	STA/AP Source Constant: 0 = long preamble
FCB1	cnfExcludeLongPreamble	AP Source Constant: False
FCB3	cnfBasicRates	STA/AP Source Constant: 3 = 11Mbps

Table 4.19: Network Parameters, Static Configuration Entities – Initial Values (Continued)

RID	Name	Initial Value
FCB4	cnfSupportedRates	STA/AP Source Constant: 15
FCBA	CnfPassiveScanCtrl	STA Source Constant: 0

Table 4.20 lists the initial values for Dynamic Configuration entities.

Table 4.20: Network Parameters, Dynamic Configuration Entities – Initial Values

RID	Name	Initial Value
FC09	cnfPMEnabled	STA Source constant: 0 = No
FC0A	cnfPMEPS	STA Source constant: 1 = Yes
FC0D	cnfPMHoldoverDuration	STA Source constant: 100
FC23	cnfWEPDefaultKeyID	STA/AP Source Constant
FC24	cnfDefaultKey0	STA/AP Source Constant: not valid until written
FC25	cnfDefaultKey1	STA/AP Source Constant: not valid until written
FC26	cnfDefaultKey2	STA/AP Source Constant: not valid until written
FC27	cnfDefaultKey3	STA/AP Source Constant: not valid until written
FC28	cnfWEPFlags	STA/AP Source Constant: not valid until written
FC29	cnfWEPKeyMappingTable	STA/AP Source Constant: not valid until written
FC2A	cnfAuthentication	STA/AP Source Constant: 0 = Open Systems
FC32	cnfAltRetryCnt	STA/AP Source Constant: 7
FC33	cnfBeaconInterval	AP Source Constant: 100mSec
FC37	cnfPriorityQUsage	STA/AP Source Constant: True (0x000F)
FC80	GroupAddresses	STA Source constant: RecordLen: 1
FC81	createIBSS	STA Source Constant: 0
FC82	FragmentationThreshold	STA Source constant: 2346
FC83	RTSThreshold	STA Source constant: 2432
FC84	TxRateControl	STA Source constant: 3 = Auto Fallback
FC85	PromiscuousMode	STA Source constant: 0 = No
FC90	FragmentationThreshold0	AP Source constant: 2346
FC91	FragmentationThreshold1	AP Source constant: 2346
FC92	FragmentationThreshold2	AP Source constant: 2346
FC93	FragmentationThreshold3	AP Source constant: 2346
FC94	FragmentationThreshold4	AP Source constant: 2346
FC95	FragmentationThreshold5	AP Source constant: 2346
FC96	FragmentationThreshold6	AP Source constant: 2346
FC97	RTSThreshold0	AP Source constant: 2432
FC98	RTSThreshold1	AP Source constant: 2432
FC99	RTSThreshold2	AP Source constant: 2432

Table 4.20: Network Parameters, Dynamic Configuration Entities – Initial Values (Continued)

RID	Name	Initial Value
FC9A	RTSThreshold3	AP Source constant: 2432
FC9B	RTSThreshold4	AP Source constant: 2432
FC9C	RTSThreshold5	AP Source constant: 2432
FC9D	RTSThreshold6	AP Source constant: 2432
FC9E	TxRateControl0	AP Source constant: 3 = Auto Fallback
FC9F	TxRateControl1	AP Source constant: 3 = Auto Fallback
FCA0	TxRateControl2	AP Source constant: 3 = Auto Fallback
FCA1	TxRateControl3	AP Source constant: 3 = Auto Fallback
FCA2	TxRateControl4	AP Source constant: 3 = Auto Fallback
FCA3	TxRateControl5	AP Source constant: 3 = Auto Fallback
FCA4	TxRateControl6	AP Source constant: 3 = Auto Fallback
FCB5	cnfFallbackCtrl	Source constant: 0x047C
FCB6	WEPKeyDisable	Source constant: 0
FCB7	WEPKeyMapIndex	AP Source constant: 0xFF
FCB8	BroadcastKeyID	AP Source constant: 2
FCB9	EntSecFlag	AP Source constant: 0

Table 4.21 lists the initial values for Behavior Parameters.

Table 4.21: Behavior Parameters, Initial Values

RID	Name	Initial Value
FCE0	TickTime	PRI Source constant: 10

4.4.4.2 Information Records Initial Values

Information Record values, for which a Dynamic origin is specified, are obtained from that origin record when the Access / read command is performed.

Table 4.22 lists the initial value and dynamic origin for NIC Information Records.

Table 4.22: NIC Information Records – Initial Values

RID	Name	Initial Value	Dynamic Origin
FD00	MaxLoadTime	PRI Source constant (depends on Flash Memory type): 5000	None
FD01	DownloadBuffer	PRI Source constants (depend on Memory Map): 0060 hex → DownloadBufferPage 0000 hex → DownloadBufferOffset 0200 hex → DownloadBufferLength	None
FD02	PRIIdentity	PRI Source constants: 21 → PRICompID 1 → PRIVariant 1 → PRIMajorVersion PRIMinorVersion: Impl. value	None

Table 4.22: NIC Information Records – Initial Values (Continued)

RID	Name	Initial Value	Dynamic Origin
FD03	PRISupRange	PRI Source constants: 0 = Off → PRIRole 3 → PRIID 1 → PRIVariant 1 → PRIBottom 1 → PRITop	None
FD04	CFIActRanges	PRI Source constants: 1 = On → CFIRole 2 → CFIID 1 → CFIVariant 1 → CFIBottom 1 → CFITop	None
FD0A	NICSerialNumber	PDR 0003: NIC Serial Number PRI Source constant: “90UT01000000”	None
FD0B	NICIdentity	PDR 0008: NIC Component Identification PRI Source constants: 1 → PRICompID 1 → PRIVariant 1 → PRIMajorVersion 0 → PRIMinorVersion: Impl. value	None
FD0C	MFISupRange	PDR 0006: RF Modem Interface Compatibility Levels PRI Source constants: 0 = Off → PRIRole 1 → PRIID 1 → PRIVariant 1 → PRIBottom 1 → PRITop	None
FD0D	CFISupRange	PDR 0007: PRISM MAC Interface Compatibility Levels PRI Source constants 0 = Off → PRIRole 2 → PRIID 1 → PRIVariant 1 → PRIBottom 1 → PRITop	None
FD10	ChannelList	PDR 0104: Allowed Channel Bitmap STA and AP Source constants: 1FFF hex	None
FD11	RegulatoryDomains	PDR 0103: Regulatory Domain List STA and AP Source constants: StringLen = 2 Byte Area = 30, 31 hex	None
FD12	TempType	PDR 0107: Temperature Type STA and AP Source constants: 0001 hex	None
FD13	CIS	Compact CIS Area	None

Table 4.22: NIC Information Records – Initial Values (Continued)

RID	Name	Initial Value	Dynamic Origin
FD20	STAIdentity	STA Source constants: 31 → STACompID 1 → STAVariant 1 → STAMajorVersion STAMinorVersion: impl. value	None
FD21	STASupRange	STA Source constants: 0 = Off → STARole 4 → STAID 1 → STAVariant 1 → STABottom 1 → STATop	None
FD22	MFIActRanges	STA Source constants: 1 = On → MFIRole 1 → MFIID 1 → MFIVariant 1 → MFIBottom 1 → MFITop	None
FD23	CFIActRanges	STA Source constants: 1 = On → CFIRole 2 → CFIID 1 → CFIVariant 1 → CFIBottom 1 → CFITop	None
FFFE	BuildSequence	Firmware Build Sequence Code: 0 = Released build not 0 = In-house build	
FFFF	FWId	From Host.	

Table 4.23 lists the initial value and dynamic origin for MAC Information Records.

Table 4.23: MAC Information Records – Initial Values

RID	Name	Initial Value	Dynamic Origin
FD40	PortStatus	STA Source constant: 1 = Disabled	PortStatus variable
FD41	CurrentSSID	STA Source constant: StringLen = 0	SSID variable
FD42	CurrentBSSID	STA Source constant: 00-00-00-00-00-00 hex	BSSID variable
FD43	CommsQuality	STA Source constants: 0 → CQ.currBSS 27 → ASL.currBSS 27 → ANL.currFC	CQ.currBSS variable ASL.currBSS variable ANL.currFC variable
FD44	CurrentTxRate	STA Source constant: 2	CurrentTxRate variable
FD45	CurrentBeaconInterval	STA and AP Fixed Configuration: OwnBeaconInterval: 100	CurrentBeaconInterval variable

Table 4.23: MAC Information Records – Initial Values (Continued)

RID	Name	Initial Value	Dynamic Origin
FD46	CurrentScaleThresholds	STA Fixed Configuration: StaLowScaleThresholds: 48, 48, 48, 20, 12 AP Fixed Configuration: StaLowScaleThresholds: 48, 48, 48	CurrentScaleThresholds variable
FD47	ProtocolRspTime	STA and AP Fixed Configuration: ProtocolRspTime: 100	None
FD48	ShortRetryLimit	STA and AP Fixed Configuration: ShortRetryLimit: 7	None
FD49	LongRetryLimit	STA and AP Fixed Configuration: LongRetryLimit: 4	None
FD4A	MaxTransmitLifetime	STA and AP Fixed Configuration: MaxTransmitLifetime: 512	None
FD4B	MaxReceiveLifetime	STA and AP Fixed Configuration: MaxReceiveLifetime: 512	None
FD4C	CFPollable	STA Source constant: 0 = No	None
FD4D	AuthenticationAlgorithms	STA and AP Source constants: 1 → AuthenticationType 1 = Yes → TypeEnabled	None
FD4F	PrivacyOptionImplemented	STA and AP Source constant: 0 = No	None
FD80	CurrentTxRate1	AP Source constant: 2	CurrentTxRate1 variable
FD81	CurrentTxRate2	AP Source constant: 2	CurrentTxRate2 variable
FD82	CurrentTxRate3	AP Source constant: 2	CurrentTxRate3 variable
FD83	CurrentTxRate4	AP Source constant: 2	CurrentTxRate4 variable
FD84	CurrentTxRate5	AP Source constant: 2	CurrentTxRate5 variable
FD85	CurrentTxRate6	AP Source constant: 2	CurrentTxRate6 variable
FD86	OwnMACAddress	PDR 0101: Universal MAC Address AP Source constant: 00-00-00-00-00-00 hex	None
FD88	ScanResult	PDR 0101: Universal MAC Address	

Table 4.24 lists the initial value and dynamic origin for Modem Information Records.

Table 4.24: Modem Information Records – Initial Values

RID	Name	Initial Origin	Dynamic Origin
FDC0	PHYType	STA and AP Source constant: 2 = dsss2dot4GHz	None
FDC1	CurrentChannel	PDR 0105: Default channel number STA and AP Source constant: 3	CurrentChannel variable
FDC2	CurrentPowerState	STA and AP Source constant: 2 = Doze	ModemState variable

Table 4.24: Modem Information Records – Initial Values (Continued)

FDC3	CCAMode	STA and AP Source constant: 2 = Carrier sense only	None
FDC6	SupportedDataRates	STA and AP Source constants: StringLen = 2 Byte Area = 2, 4	None

4.4.5 INFORMATION FRAME STRUCTURE FORMATS

Three sets of Information Frame Structures are used to pass communications-related information between Host software and the PRISM MAC:

- *Notification Frame Structures* are used to provide information by means of the Notify command through which the Host software adapts internal PRISM MAC behavior to specific communications needs.
- *Inquiry Frame Structures* are used to obtain information that is gathered as requested by the Host software through an Inquire command.
- *Unsolicited Frame Structures* are used to obtain information that is collected by the asynchronous communications processes and issued to the Host software if specific internal conditions are met.

Notice that, internally, a structure buffer must be allocated for an Unsolicited Frame Structure. If not enough buffer space is available, an InfDrop event is generated (see 4.3.1.2 Frame Structure Storage Limit Handling).

Notice that the Info events related to Inquiry Frames and Unsolicited Frames must always be acknowledged. Otherwise, these frames remain in On-board memory, which may cause memory exhaustion.

Alert: A dagger symbol (†) in the RID identification subsection label indicates that the RID is in development and is not fully implemented at the time of publication of this document.

The general format for Information Frame Structures exchanged between the Host software and the PRISM MAC is:

Word Offset	Field name	Size (words)
0	FrameLen: 1–1000	1
1	InfoType: **F0xx–F2xx**	1
2	Data (optional)	

The FrameLen value is an Integer that indicates the number of words of the remainder of the structure, and the InfoType value is a Byte Area with a hexadecimal contents.

The specific frame structure descriptions in the following sections define the Data fields format by means of a Format Descriptor. These descriptors are defined in 4.3.2 Communications Frame Structure Formats.

4.4.5.1 Notification Frames, MAC Management

MAC Management Notification Frames contain information regarding the relationship of the local node with other network components. This information influences the behavior of the asynchronous communications processes, and allows the AP Host to notify the AP firmware that a station has reassociated with another AP. The AP firmware will clean up internal data structures associated with the station.

4.4.5.1.1 HandoverAddress (AP Only)

Word Offset	Field name	Size (words)
0	FrameLen: 4	1
1	InfoType: **F000**	1
2	HandoverAddress	3

Field Descriptions

Name: HandoverAddress

Description: Specifies the MAC Address of a Station that has performed an AP-only reassociation with another Access Point.

Format: Byte Area

4.4.5.2 Inquiry Frames, Diagnose

Diagnose Inquiry frames contain management- and diagnostics-related values gathered by the asynchronous communications processes after an Inquire command is issued (assuming sufficient buffer space is available; see 4.2 Interface Commands).

The following Inquiry InfoTypes are assigned:

Table 4.25: Diagnose Inquiry Frames

Diagnose		
InfoType	**Information Name**	**Description**
F100	CommunicationTallies	Communications-related statistical counter values.
F101	ScanResults	Information from network connection points within range.
F102	ChannelInfoResults	Bit-map of channels scanned during current sweep (AP Only).
F103	HostScanResults	Information returned from host-initiated non-destructive scan.

Note that CommunicationTallies can also be generated unsolicited.

4.4.5.2.1 CommunicationTallies

cnfThirty2Tally (RID FC42) can be set to support either 16-bit or 32-bit tally counts (see 4.4.2.1.23 cnfThirty2Tally). If this RID is set to its default value of False, tallies are 16-bit values, and the Information frame returned is as follows:

Word Offset	Field name	Size (words)
0	FrameLen: 22	1
1	InfoType: **F100**	1
2	TxUnicastFrames	1
3	TxMulticastFrames	1
4	TxFragments	1
5	TxUnicastOctets	1
6	TxMulticastOctets	1
7	TxDeferredTransmissions	1
8	TxSingleRetryFrames	1
9	TxMultipleRetryFrames	1
10	TxRetryLimitExceeded	1
11	TxDiscards	1
12	RxUnicastFrames	1
13	RxMulticastFrames	1
14	RxFragments	1
15	RxUnicastOctets	1
16	RxMulticastOctets	1
17	RxFCSErrors	1
18	RxDiscardsNoBuffer	1
19	TxDiscardsWrongSA	1
20	RxDiscardsWEPUndecryptable	1
21	RxMessageInMsgFragments	1
22	RxMessageInBadMsgFragments	1

With cnfThirty2Tally set to False, tally fields contain an integer in the following range:

0–65534 decimal = Supported tally count
FFFF hexadecimal = Tally not supported.

If any tally reaches the value of 59904 decimal (0xEA00 hexadecimal), the firmware generates an unsolicited Information frame and clears all tallies.

If any tally reaches the value of 65534 decimal (0xFFFE hexadecimal), the firmware clamps the value (will no longer increment it), and generates an InfoDrop event in the Event Status register.

If cnfThirty2Tally is set to True, tallies are 32-bit values, and the Information frame returned is as follows:

Word Offset	Field name	Size (words)
0	FrameLen: 42	1
1	InfoType: **F100**	1
2	TxUnicastFrames	2
4	TxMulticastFrames	2
6	TxFragments	2
8	TxUnicastOctets	2
10	TxMulticastOctets	2
12	TxDeferredTransmissions	2
14	TxSingleRetryFrames	2
16	TxMultipleRetryFrames	2
18	TxRetryLimitExceeded	2
20	TxDiscards	2
22	RxUnicastFrames	2
24	RxMulticastFrames	2
26	RxFragments	2
28	RxUnicastOctets	2
30	RxMulticastOctets	2
32	RxFCSErrors	2
34	RxDiscardsNoBuffer	2
36	TxDiscardsWrongSA	2
38	RxDiscardsWEPUndecryptable	2
40	RxMessageInMsgFragments	2
42	RxMessageInBadMsgFragments	2

With cnfThirty2Tally set to True, tally fields contain an integer in the following range:

0–4294967294 decimal = Supported tally count
FFFFFFFF hexadecimal = Tally not supported.

If the tally reaches the value of 4294961664 decimal (0xFFFFEA00 hexadecimal), the firmware generates an unsolicited Information frame and clears all tallies.

If any tally reaches the value of 4294967294 decimal (0xFFFFFFFF hexadecimal), the firmware clamps the value (will no longer increment it), and generates an InfoDrop event in the Event Status register.

For both 16-bit and 32-bit tallies, each tally field gives a count obtained from a related internal counter. These internal counters are reset to zero after the values are copied to this Information frame.

NOTE: Since tallies are maintained before the multicast filtering (according to the configured GroupAddresses) takes place, the appropriate tallies include the MSDUs that are discarded because of the multicast filter.

Field Descriptions

Name: Tal.TxUnicast (Unicast Frames Transmitted)

Description: The total number of successfully transmitted MSDUs for which the Destination Address is a unicast MAC address. This implies having received an acknowledgment to all associated MPDUs.

Name: Tal.TxMulticast (Multicast Frames Transmitted)

Description: The total number of successfully transmitted MSDUs for which the Destination Address is a multicast MAC address (including the broadcast MAC address). When operating as a station in an ESS, these frames are directed to the AP. This implies having received an acknowledgment to all associated MPDUs.

Name: Tal.TxFrags (Fragments Transmitted)

Description: The total number of successfully delivered MPDUs of type Data or Management. This includes directed MPDUs transmitted and being ACKed, as well as non-directed MPDUs transmitted.

Name: Tal.TxUniOctets (Unicast Octets Transmitted)

Description: The total number of octets transmitted successfully as part of successfully transmitted unicast MSDUs (see Tal.TxUnicast). These octets include the MAC Header and Frame Body [std 7.1.2] of all associated fragments.

Name: Tal.TxMultiOctets (Multicast Octets Transmitted)

Description: The total number of octets transmitted successfully as part of successfully transmitted multicast (including broadcast) MSDUs (see Tal.TxMulticast). These octets include the MAC Header and Frame Body [std 7.1.2] of all associated fragments.

Name: Tal.TxDeferredTx (Deferred Transmissions)

Description: The number of MSDUs for which one or more (or a fragment of one or more) transmission attempt(s) was deferred to avoid a collision.

Name: Tal.TxSingleRetries (Single Retry Frames Transmitted)

Description: The number of MSDUs successfully transmitted after one (and only one) retransmission (on the total of all associated fragments).

Name: Tal.TxMultipleRetries (Multiple Retry Frames Transmitted)

Description: The number of MSDUs successfully transmitted after more than one retransmission (on the total of all associated fragments).

Name: Tal.TxRetryLimitExc (Transmit Retry Limit Exceeded)

Description: The number of times a MSDU was not transmitted successfully because the retry limit (either the ShortRetryLimit or the LongRetryLimit) was reached, due to no acknowledgment or CTS received.

Name: Tal.TxDiscards (Discarded Transmissions)

Description: The number of transmit requests that were discarded to free up buffer space on the NIC. This tally is incremented when one of the following occurs:

- Transmit queued too long on one of the transmit queues, due to many retries and defers, or otherwise not being able to transmit (for example, scanning).
- Transmit queued too long on the Power-Save queue (for a station in IBSS, the destination station did not respond to ATIM; for an AP, the station did not Poll or wake up in time).

Name: Tal.RxUnicast (Unicast Frames Received)

Description: The total number of successfully received MSDUs with a unicast MAC address as the Destination Address.

Name: Tal.RxMulticast (Multicast Frames Received)

Description: The total number of successfully received MSDUs with a multicast MAC address (including the broadcast MAC address) as the Destination Address.

Name: Tal.RxFrags (Fragments Received)

Description: The total number of successfully received MPDUs of type Data or Management.

Name: Tal.RxUniOctets (Unicast Octets Received)

Description: The total number of octets received successfully as part of unicast MSDUs (see Tal.RxUnicast). These octets include the MAC Header and Frame Body [std 7.1.2] of all associated fragments.

Name: Tal.RxMultiOctets (Multicast Octets Received)

Description: The total number of octets received successfully as part of multicast (including broadcast) MSDUs (see Tal.TxMulticast). These octets include the MAC Header and Frame Body [std 7.1.2] of all associated fragments.

Name: Tal.RxFCSErrors (FCS Errors Received)

Description: The number of MPDUs considered to be destined for this station (Address1 matches) received with an FCS error. This does not include “things” received with an incorrect CRC in the PLCP header. These are not considered MPDUs.

Name: Tal.RxNoBuffer (Receives Discarded - No Buffer)

Description: The number of received MPDUs that were discarded because of a lack of buffer space on the NIC.

Name: Tal.TxWrongSA (Receives Discarded - Wrong SA)

Description: The number of transmit requests that were discarded because of wrong Source Address (Source Address ≠ OwnMACAddress). This only applies to a station with a BSS Port.

Name: Tal.RxWEPUndecryp (Receives Discarded - WEP Undecryptable)

Description: The number of received MPDUs, with the WEP subfield in the Frame Control field set to one, that were discarded because either it should not have been encrypted, or due to the receiving station not implementing the privacy option.

Name: Tal.RxMsgInMsg (Messages Received in Message Fragments)

Description: The total number of MPDUs of type Data or Management received successfully, while there was another good reception going on above the carrier detect threshold (the message-in-message path #1 in the RF modem).

Name: Tal.RxMsgInBadMsg (Messages Received in Bad Message Fragments)

Description: The total number of MPDUs of type Data or Management received successfully, while there was another reception going on above the carrier detect threshold but with bad or incomplete PLCP Preamble and Header (the message-in-message path #2 in the RF modem).

4.4.5.2.2 ScanResults (Station Only)

Word Offset	Field name	Size (words)
0	FrameLen: 3+*n**31, where *n*=0–32	1
1	InfoType: **F101**	1
2	Reserved for future use	1
3	Scan Reason	1
4	Scan Result 1 (optional)	31
	...	
35 + (*n*-1) x 31	Scan Result *n* (optional)	31

Field Descriptions

Name: Scan Reason

Description: The reason this scan was initiated, where

1 = Host initiated
2 = Firmware initiated due to communications quality level
3 = Inquiry request from host for current SRT

Name: ScanResult *N*

Description: Each Scan Result field (when present) gives the scan result information for one Basic Service Set (BSS) (Station Only).

The actual number of Scan Result fields depends on the number of BSSs detected during the Scan. The Scan Result fields are sorted as follows:

- Extended Service Set (ESS) Scan Results come first, in descending order by signal-to-noise ratio.
- Independent Basic Service Set (IBSS) results come after the ESS results, also in descending order of signal-to-noise ratio.

The scan results are based on Probe Response frames received.

When the RoamingMode is set to “Roaming Decision Made By Host”, unsolicited Scan Results frames will be generated when the firmware performs a background scan.

The format of each Scan Result (Station only) sub-field is:

Word Offset	Field name	Size (words)
0	SRT.CHID	1
1	SRT.ANL	1
2	SRT.SL	1
3	SRT.BSSID	3
6	SRT.BcnInt	1
7	SRT.Cap	1
8	SRT.SSID	17
25	SRT.SupRates	5
30	SRT.Rate	1

Sub-field Descriptions

Name: SRT.CHID

Description: BSS Channel ID from Probe Response

Format: Integer

Values: Range: 1–14

Name: SRT.ANL

Description: Average noise level of the Probe Responses received during the scan, averaged on a per-channel basis.

Format: Integer

Values: Range: 001B–009A hexadecimal

Name: SRT.SL

Description: Signal level on which the Probe Response was received.

Format: Integer

Values: Range: 001B–009A hexadecimal

Name: SRT.BSSID

Description: MAC address of BSS responder from Probe Response.

Format: MAC address

Values: Any

Name: SRT.BcnInt

Description: BSS Beacon Interval (in Kμs) from Probe Response.

Format: Integer

Values: Any

Name: SRT.Cap

Description: BSS Capability information, copied from the Probe Response.

Format: Bit-mapped integer

Values: See Section 7.3.1.4 of Reference [1].

Name: SRT.SSID

Description: Copy of the Service Set Identifier element from the Probe Response. That is, the first word contains the length (the element type is 0) in the high-order octet, followed by 0 to 32 octets that contain the SSID string itself.

Format: String

Values: Length word, followed by string of length 0–32

Name: SRT.SupRates

Description: Copy of the contents Supported Rates element from the Probe Response. This subfield is terminated with a zero stamp word. Each data rate is encoded as defined in Section 7.3.2.2 of Reference [1].

Format: Byte string

Values: Any

Name: SRT.Rate

Description: Data rate of the received Probe Response. The data rate is encoded as defined in Section 18.2.3.3 of Reference [7], where:

0x0A = 1 Mb/s
0x14 = 2 Mb/s
0x37 = 5.5 Mb/s
0x6E = 11 Mb/s

Format: Byte string

Values: Any

4.4.5.2.3 ChannelInfoResults (AP Only)

Word Offset	Field name	Size (words)
0	FrameLen: 2+*n**4, where *n*=0–16	1
1	InfoType: **F102**	1
2	Scan Channels	1
3	Channel Result 1 (optional)	4
	...	
7 + 4(*n*-1)	Channel Result *n* (optional)	4

Field Descriptions

Name: Scan Channels

Description: Bit-mask of the channels that were scanned during this sweep. See 4.4.2.2.28 ChannelInfoRequest (AP Only)) for a description of the mapping between channels and bits.

Format: Bit-mapped integer

Values: any (reserved bits 0)

Name: Channel Result *N*

Description: Each Channel Result field (when present) gives the activity level information for that channel.

The format of each Channel Result sub-field is:

Word Offset	Field name	Size (words)
0	CRT.CHID	1
1	CRT.ANL / CRT.ASL	1
2	CRT.PNL / CRT.PSL	1
3	CRT.Activity	1

Sub-field Descriptions

Name: CRT.CHID

Description: BSS Channel ID from Probe Response.

Format: Integer

Values: Range: 1–14

Name: CRT.ANL/CRT.ASL

Description: Average noise level of the channel during the scan, normalized to dB values. If there is 802.11 network activity detected in the channel, this value will represent the average signal level of received beacons.

Format: Integer

Values: Range: True dBm

Name: CRT.PNL/CRT.PSL

Description: Peak noise level of the channel during the scan, or (if 802.11 network activity was detected in the channel) the peak signal level. This value is calculated using the RSSI.

Format: Integer

Values: Range: True dBm

Name: CRT.Activity

Description: This value represents the type of 802.11 activity found and whether an 802.11 PCF is active.

Bit	Activity
0	if true, indicates 802.11 is active
1	if true, indicates 802.11 PCF is active

Format: Boolean Flag

Values: 0 or 1 for each field

4.4.5.2.4 HostScanResults (Station Only)

The HostScanResults information frame is returned by the firmware after RID FCE5 (see 4.4.2.2.29 HostScan) is invoked.

Word Offset	Field name	Size (words)
0	FrameLen: 3+*n**32, where *n*=0–32	1
1	InfoType: **F103**	1
2	resultSize	1
3	reserved	1
4	Host Scan Result 1 (optional)	31
	...	
35 + (*n*-1) x 31	Host Scan Result *n* (optional)	31

Field Descriptions

Name: resultSize

Description: The size of each Host Scan Result entry. Host software uses this field as an offset into the table for the next entry.

Format: Integer

Values: 31

Name: Host ScanResult *N*

Description: Each Host Scan Result field (when present) gives the scan result information for one Basic Service Set (BSS).

The actual number of Host Scan Result fields depends on the number of BSSs detected during the Scan. The Host Scan Result fields are not sorted.

The scan results are based on Probe Response frames received.

When the RoamingMode is set to “Roaming Decision Made By Host”, unsolicited Scan Results frames will be generated when the firmware performs a background scan.

The format of each Host Scan Result sub-field is as follows:

Word Offset	Field name	Size (words)
0	HSR.CHID	1
1	HSR.ANL	1
2	HSR.SL	1
3	HSR.BSSID	3
6	HSR.BcnInt	1
7	HSR.Cap	1
8	HSR.SSID	17
25	HSR.SupRates	5
30	HSR.Rate	1
31	HSR.ATIM	1

Sub-field Descriptions

Name: HSR.CHID

Description: BSS Channel ID from Probe Response

Format: Integer

Values: Range: 1–14

Name: HSR.ANL

Description: Average noise level of the Probe Responses received during the scan, averaged on a per-channel basis. This value is then normalized to a dBm value.

Format: Integer

Values: Range, based on the environment in which the radio is operating. In a quiet environment the value will be very low (approximately -96).

Name: HSR.SL

Description: Signal level on which the Probe Response was received, normalized to a dBm value.

Format: Integer

Values: Range, based on the range of the radio from the source and the sensitivity of the radio. Values are typically between -20 and -90.

Name: HSR.BSSID

Description: MAC address of BSS responder from Probe Response.

Format: MAC address

Values: Any

Name: HSR.BcnInt

Description: BSS Beacon Interval (in Kμs) from Probe Response.

Format: Integer

Values: Any

Name: HSR.Cap

Description: BSS Capability information, copied from the Probe Response.

Format: Bit-mapped integer

Values: See Section 7.3.1.4 of Reference [1].

Name: HSR.SSID

Description: Copy of the Service Set Identifier element from the Probe Response. The first word contains the length (the element type is 0) in the high-order octet, followed by 0 to 32 octets that contain the SSID string itself.

Format: String

Values: Length word, followed by string of length 0–32

Name: HSR.SupRates

Description: Copy of the contents Supported Rates element from the Probe Response. This sub-field is terminated with a zero stamp word. Each data rate is encoded as defined in Section 7.3.2.2 of Reference [1].

Format: Byte string

Values: Any

Name: HSR.Rate

Description: Data rate of the received Probe Response. The data rate is encoded as defined in Section 18.2.3.3 of Reference [7], where:

0x0A = 1 Mb/s
0x14 = 2 Mb/s
0x37 = 5.5 Mb/s
0x6E = 11 Mb/s

Format: Byte string

Values: Any

Name: HSR.ATIM

Description: The ATIM window time (in Kµs) used by the IBSS network.

Format: Integer

Values: Any

4.4.5.3 Unsolicited Frames, Diagnose

Diagnose Unsolicited frames contain management- and diagnostics-related values collected by the asynchronous communications processes.

4.4.5.3.1 CommunicationTallies

If one or more of the internal counters related to the CommunicationTallies reaches a value that is higher than 59904 decimal (i.e., EA00 hexadecimal), a CommunicationTallies frame is generated and all related internal counters are reset to zero after the values are copied to this Information frame.

The frame format is the same as the CommunicationTallies Inquiry frame format (see 4.4.5.2.1 CommunicationTallies).

If not enough buffer space is available (and an InfDrop event is generated) (see 4.3.1.2 Frame Structure Storage Limit Handling), the internal counters are not reset to zero. Hence, a next attempt to generate the frame is performed at the next tally increment that results in a value higher than 59904 decimal.

If one or more tallies reach the maximum value of 65534 decimal (i.e., FFFE hexadecimal), these tallies keep this maximum value until an attempt to generate the frame is successful.

Note that CommunicationTallies may also be requested by means of the Inquire command (see 4.2.3.2 Inquire Command).

4.4.5.4 Unsolicited Frames, MAC Management

MAC Management Unsolicited frames contain information collected by the asynchronous communications processes regarding the relationship of the local node with other network components.

If not enough buffer space is available (and an InfDrop event is generated) (see 4.3.1.2 Frame Structure Storage Limit Handling), no other attempt is made and the information is dropped.

The following MAC Management Unsolicited InfoTypes are assigned:

MAC Management		
InfoType	**Information Name**	**Description**
F200	LinkStatus	Station connection change.
F201	AssociationStatus	[AP] association change related to a specific station.
F202	AuthenticationRequest	[AP] authentication request from a specific station.
F203	PowerSaveUserCount	[AP] generated when the number of PS stations changes.
F204	KeyIdChanged	[AP] generated when key-mapped station changes keys

4.4.5.4.1 LinkStatus

Word Offset	Field name	Size (words)
0	FrameLen: 2	1
1	InfoType: **F200**	1
2	LinkStatus	1

Field Descriptions

Name: LinkStatus

Description: Indicates a change of the connection control status as maintained within a Station.

Format: Integer, as follows:

1 = Connected
2 = Disconnected
3 = Access Point Change
4 = Access Point Out Of Range
5 = Access Point In Range
6 = Association failed

Following a Join Request, a LinkStatus of 3 or 1 indicates that the Join Request was successful. A value of 6 indicates that the Join Request was unsuccessful.

The Station maintains connection with an AP when an AP out-of-range status is sent. This message is a notification to the Host so that the Host may suspend transmit operations until the Station moves back into range.

Range: 1–6 decimal.

4.4.5.4.2 AssociationStatus

Word Offset	Field name	Size (words)
0	FrameLen: 10	1
1	InfoType: **F201**	1
2	Association Status	1
3	Station Address	3
6	Old AP Address (Optional)	3
9	Reason	1
10	Reserved	1

Field Descriptions

Name: AssociationStatus

Description: Indicates a change of the association status of a specific Station as maintained by an Access Point.

Format: Integer, as follows:

1 = STA Associated
2 = STA Reassociated
3 = STA Disassociated
4 = Association Failure
5 = Authentication Failure

Range: 1–5 decimal.

Name: StationAddress

Description: MAC address of the Station concerned.

Format: Byte Area.

Values: Any

Name: OldAPAddress

Description: MAC Address of the AP with which the station was previously associated. This field is valid only if Association Status = 2 (Reassociated).

Format: Byte Area.

Values: Any

Name: Reason

Description: If Association Status = 3 (Disassociated), 4 (Association Failed), or 5 (Authentication Failed), this field will contain a code that indicates why the station was disassociated or the association attempt failed.

Format: Integer.

Values: These reason codes are found in Reference [1].

4.4.5.4.3 AuthenticationRequest (AP Only)

Word Offset	Field name	Size (words)
0	RecordLen: 5	1
1	Info Type: **F202**	1
2	Station Address	3
5	Authentication Algorithm Number	1

The AuthenticationRequest notification will only be generated if the Host had previously set cnfHostAuthentication (see 4.4.2.1.18 cnfHostAuthentication (AP Only)) to true. When the Host receives an AuthenticationRequest notification, it sends an authenticateStation RID to allow the firmware to complete the authentication sequence.

Field Descriptions

Name: Station Address

Description: MAC Address of station requesting authentication.

Format: MAC Address

Values: Any

Name: Authentication Algorithm Number

Description: The authentication algorithm number that the station is requesting to use.

Format: Integer

Values: One of the following:

0 – 802.11 Open System Authentication
1 – 802.11 Shared Key Authentication

4.4.5.4.4 PowerSaveUserCount (AP only)

Word Offset	Field name	Size (words)
0	RecordLen: 2	1
1	Info Type: **F203**	1
2	UserCount	1

Field Descriptions

Name: UserCount

Description: The number of stations currently in power save mode. This notification is automatically generated if the count transitions from 0 to 1 or 1 to 0. This notification allows the host to perform different queuing models based on the current power save state of the associated stations.

Format: Integer

Values: Range: 0–n; where n is the number of associated stations.

4.4.5.4.5 KeyIdChanged (AP Only)

Word Offset	Field Name	Size (words)
0	RecordLen: 5	1
1	Info Type: F204	1
2	STA Address	3
5	Key ID	1

Field descriptions

Name: STA Address

Description: 48-bit MAC address of the Station whose Key ID has changed.

Format: 48-bit IEEE 802 MAC Individual address.

Values: Any

Name: Key ID

Description: WEP Key Identifier

Format: Byte

Values: 0 or 1

After WEP Key-Mapping has been enabled (see 4.4.2.2.7 cnfWEPKeyMapping (AP Only)), the firmware will begin sending unsolicited KeyIdChanged info frames to the host interface for notification purposes. This occur each time the Key ID field within a received frame differs in value from the Key ID field received in previous frames. This mechanism will allow the host software to know when the Station has started using a newly specified WEP Key.

4.4.6 FRAME EXCHANGE SEQUENCES

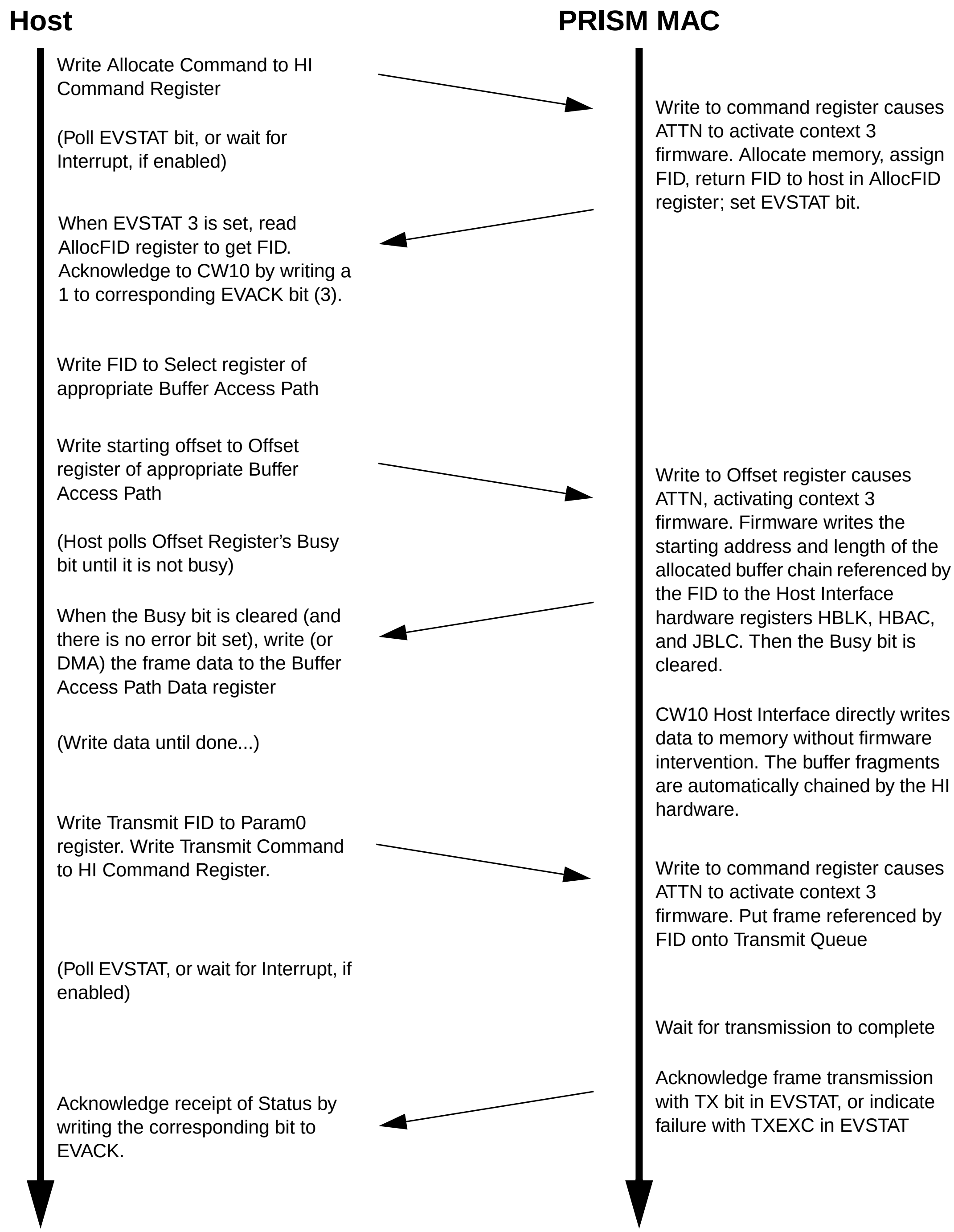


Figure 4.2: Sending a Frame to the PRISM MAC

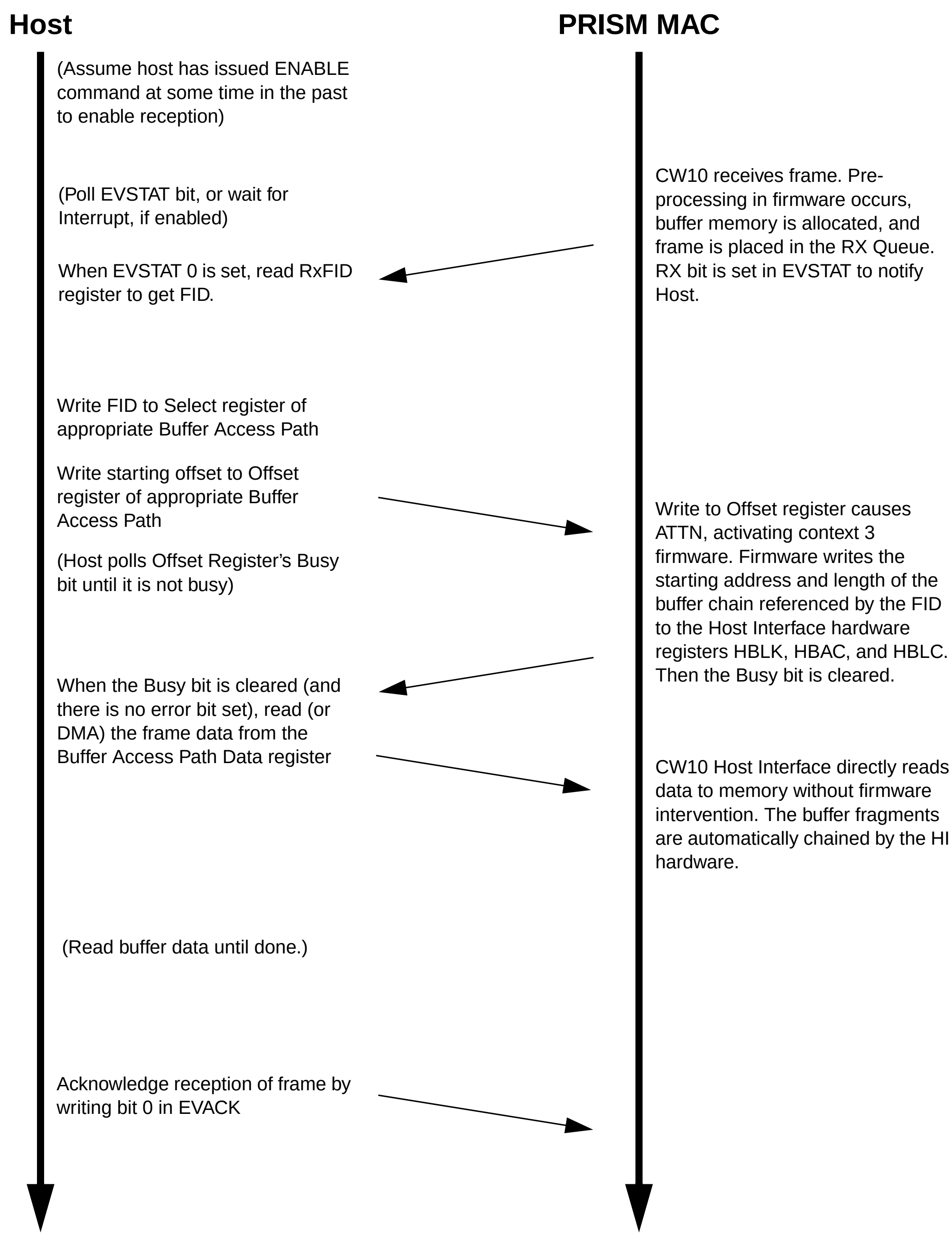


Figure 4.3: Receiving a Frame from the PRISM MAC

Chapter 5 Universal Serial Bus (USB) Host Interface APIs and Implementation (3842 Specific)

5.1 Overview

The USB interface is implemented using the Phoenix USB Device Controller (UDC) core and logic to connect the UDC core to the HFA3842. See Reference [8] for more information on the USB protocol.

The HFA3842 presents five USB endpoints.

- Endpoint0, Control IN, 8-bytes max packet, Tx0Que
- Endpoint0, Control OUT, 8-bytes max packet, Rx0Que
- Endpoint1, Bulk IN, 64-bytes max packet, Tx1Que and BAP0 (into host)
- Endpoint2, Bulk OUT, 64-bytes max packet, Rx1Que and BAP1 (out from host)
- Endpoint3, Interrupt IN, 8-bytes max packet, Tx2Que

API communications paths use the Bulk IN/OUT pipes. The Control pipe is used for enumeration and the Enable Bulk In command that must be performed just after the USB Set Configuration is completed.

The most significant difference from the PCMCIA interface is the elimination of frame descriptors (FDs) and Command Status/ACK manipulation by the Host.

5.2 Command Interface

The PCMCIA interface to the firmware relies on I/O registers and Frame Descriptors (FDs) to exchange data and to send commands. Attempts to directly map this interface to the facilities available on the USB interface are unworkable and inefficient due to the inherent limitations of USB with synchronization on 1 msec intervals. Instead, a more efficient mechanism is used to more appropriately map to the USB features.

Primary communications is accomplished using the Bulk IN/OUT endpoints to receive and send frames. Frame types include transmit and receive, commands, command response, status information, configuration data, and firmware download. The USB hardware provides automatic throttling of the data stream during times of low resources by sending NAK frames on attempts to write to the bulk pipe.

The Frame Type definitions have the following basic frame format:

Word Offset	Field name	Size (words)
0	Type	1
1	Data	0–1207

Field Descriptions

Name: Type

Description: Defines the type of frame being submitted or received.

Format: Unsigned

Values: Any

Name: Data

Description: The payload portion of the frame. The size limit of 1207 words is the total size of a maximum length packet (including the PRISM MAC header) in words.

5.3 Host-to-Firmware Frame Types

The following table defines the frame types that can be submitted to the firmware by the host and their type codes:

Frame Type	Codes
Transmit Packet	0x0000
Command Request	0x0001
Write RID	0x0002
Read RID	0x0003
Write Memory	0x0004
Read Memory	0x0005
Enable Bulk IN	0x006 (Endpoint 0 command)

5.3.1 TRANSMIT PACKET

The frame format of the transmit packet is not changed for this interface (see 4.3.2.3 Transmit Frame Control).

5.3.2 COMMAND REQUEST

The Command Request frame provides a mechanism to map the PCMCIA command interface to the USB interface. Although some commands have been replaced by the frame interface to the firmware, there is still a need to provide a generic command interface to the firmware for host software. The Command Request Frame format is:

Word Offset	Field name	Size (words)
0	Type: **0x0001**	1
1	Command	1
2	Command Parameter 0	1
3	Command Parameter 1	1
4	Command Parameter 2	1

Command Request frames will result in a Command Response frame sent by the firmware to the host. Only one command at a time can be handled by the firmware. If an additional command request is received before command handling is complete, the new command will be ignored.

5.3.3 WRITE RID

The Write RID frame provides write access to the management information base (MIB) stored in the firmware. The Write RID format is:

Word Offset	Field name	Size (words)
0	Type: **0x0002**	1
1	FrameLen	1
2	RID ID	1
3	Data	

5.3.4 READ RID

The Read RID frame provides read access to the management information base (MIB) stored in the firmware. The Read RID format is:

Word Offset	Field name	Size (words)
0	Type: **0x0003**	1
1	FrameLen	1
2	RID ID	1

5.3.5 WRITE MEMORY

The Write Memory frame provides a method to write memory on the device. The maximum length that can be written is 2048 bytes, and should not extend over page boundaries. FramLen specifies the complete length of the frame, including addresses and data.

The Write Memory frame is used with the Download Command to download a RAM resident firmware image.

Word Offset	Field name	Size (words)
0	Type: **0x0004**	1
1	FrameLen	1
2	Offset Address (low 6 bits)	1
3	Page Address (high 16 bits)	1
4	Data	

5.3.6 READ MEMORY

The Read Memory frame provides a method to access memory on the device. The maximum length that can be retrieved is 2048 bytes, and should not extend over page boundaries. FramLen specifies the complete length of the frame, including addresses and data.

Word Offset	Field name	Size (words)
0	Type: **0x0005**	1
1	FramLen	1
2	Offset Address (low 6 bits)	1
3	Page Address (high 16 bits)	1

5.4 Firmware-to-Host Frame Types

The following table defines the frame types that can be received by the host from the firmware and their type codes:

Frame Type	Codes
Packet Data	<0x7FFF
Information Frame	0x8000
Command Response	0x8001
Write RID Response	0x8002
Read RID Response	0x8003
Write Memory Response	0x8004
Read Memory Response	0x8005
Buffer Available	0x8006
USB Error	0x8007

Note that the high-order bit is zero for packet data.

5.4.1 PACKET DATA

The Packet Data Frame identifies received packets from the radio and transmits complete packets submitted by the host. Transmit complete packet frames are only reported to the host if the transmit was submitted with either the TxEx or TxOK bit set in the TxControl field (see 4.3.2.3.3 TxControl Field (word offset 6)).

The Packet Data Frame format is:

Word Offset	Field name	Size (words)
0	Type: <**0x7FFF**	1
	Data (optional)	

Field Descriptions

Name: Type

Description: Defines the type of packet being received. The bits of the type field correspond to the transmit and receive status field (as defined in 4.3.2.3 Transmit Frame Control and 4.3.2.5 Receive

Frame Control) with the exception of Bit 12 (mask: 0x1000). This bit is used to further distinguish between receive packets and transmit packets, as follows:

Bit12	Type
0	receive packet
1	transmit packet

Format: bit encoded

Values: Any

5.4.2 INFORMATION FRAME

The Information Frame format is:

Word Offset	Field name	Size (words)
0	Type: **0x8000**	1
1	FrameLen	1
2	InfoType: **F0*nn*–F2*nn***	1
3	Data (optional)	

For a description of Information Frames, refer to 4.4.5 Information Frame Structure Formats.

5.4.3 COMMAND RESPONSE

The Command Response Frame format is:

Word Offset	Field name	Size (words)
0	Type: **0x8001**	1
1	Status	1
2	Command Response 0	1
3	Command Response 1	1
4	Command Response 2	1

5.4.4 WRITE RID RESPONSE

The Write RID Response frame format is:

Word Offset	Field name	Size (words)
0	Type: **0x8002**	1
1	Status	1
2	Command Response 0	1
3	Command Response 1	1
4	Command Response 2	1

5.4.5 READ RID RESPONSE

The Read RID Response frame format is:

Word Offset	Field name	Size (words)
0	Type: **0x8003**	1
1	FrameLen:	1
2	RID ID	1
	Data (optional)	

5.4.6 WRITE MEMORY RESPONSE

The Write Memory Response frame format is:

Word Offset	Field name	Size (words)
0	Type: **0x8004**	1
1	Status	1
2	Command Response 0	1
3	Command Response 1	1
4	Command Response 2	1

5.4.7 READ MEMORY RESPONSE

The Read Memory Response frame format is:

Word Offset	Field name	Size (words)
0	Type: **0x8005**	1
1	FrameLen	1
2	Data	

5.4.8 BUFFER AVAILABLE TOKEN

This frame informs the host that there is a buffer set up and ready to receive data from the Bulk Data Out endpoint. This is used for debug purposes; the driver can disregard the Tokens. The frame format is:

Word Offset	Field name	Size (words)
0	Type: **0x8006**	1
1	FrameLen	1

5.4.9 USB ERROR INDICATION TOKEN

This frame informs the host that there has been an error in the USB Core:

Word Offset	Field name	Size (words)
0	Type: **0x8007**	1
1	ErrorType	1

5.5 Firmware Operation

The firmware maintains a queue of buffers which is used to receive frames from the Bulk OUT endpoint. The USB hardware responds with a NAK to host requests to send data to the Bulk OUT if there are no buffers ready to receive frame data.

5.5.1 PRIMARY SUPPORT

Primary firmware supports the Command Request, Write RID, Read RID, Write Memory, and Read Memory frame types.

Commands supported by primary firmware are:

- Download
- Initialize

5.5.2 SECONDARY SUPPORT

Secondary firmware supports the Transmit Packet, Command Request, Write RID, Read RID, Write Memory, and Read Memory frame types.

Commands supported by secondary firmware are:

- Initialize
- Enable
- Disable
- Inquire

5.6 Host Operation

The host maintains a posted read to the Bulk IN endpoint to receive frame data from the firmware. The USB hardware responds with a NAK to firmware requests to send data to the Bulk OUT endpoint if there are no buffers ready to receive frame data.

5.7 Download Command

The download of a firmware image uses the Command Frame interface to submit Download commands followed by the download data (see 4.2.4.2 Download Command). During a download operation, the firmware handling of the Bulk OUT endpoint is modal; command mode followed by data mode.

During command execution the PRISM MAC Controller is directed to place a firmware image into either volatile or non-volatile memory.

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	0	0	0	0	0	0	ProgMode		0	0	CmdCode = 22					

Command Controls

The programming mode value indicates the download action to be performed:

Mode	Description
0	Disable programming
1	Enable volatile memory programming
2	Enable non-volatile memory programming
3	Program non-volatile memory section

5.7.1 NON-VOLATILE PROGRAMMING

The Download command, with programming mode set to enable non-volatile memory programming (mode = 2), starts the non-volatile memory download cycle and indicates the number of bytes to be written. This number must be equal to or lower than the download buffer length as provided in the DownloadBuffer Record (see 4.4.3.1.2 DownloadBuffer).

The indicated number of bytes is written to the Bulk Data OUT endpoint following the write of the Download command. After the data is written, the firmware automatically programs the flash and the firmware is then ready to receive the next command.

Download command mode 0 completes the download process and starts running the new firmware image.

The sequence would appear as follows:

```
Send Command Frame (Download Cmd<Mode2, LowAddress, HighAddress, CodeLength>)
Send WriteMem Frame <Length, Low Address, High Address, Image Data>
Send Command Frame (Download Cmd<Mode3>)
Send Command Frame (Download Cmd<Mode2, LowAddress, HighAddress, CodeLength>)
Send WriteMem Frame <Length, Low Address, High Address, Image Data>
Send Command Frame (Download Cmd<Mode3>)
...
Send Command Frame (Download Cmd<Mode2, LowAddress, HighAddress, CodeLength>)
Send WriteMem Frame <Length, Low Address, High Address, Image Data>
Send Command Frame (Download Cmd<Mode3>)
Send Command Frame (Download Cmd<Mode0>)
```

5.7.2 VOLATILE PROGRAMMING

This command, with programming mode set to enable volatile memory programming (mode = 1), starts the volatile memory download cycle and indicates the code start address of the firmware image being downloaded.

Download command mode 0 completes the download process and starts running the new firmware image.

The sequence would appear as follows:

```
Send Command Frame (Download Cmd<Mode1,LowAddress,HighAddress, >)
Send WriteMem Frame <Length, Low Address, High Address, Image Data>
Send WriteMem Frame <Length, Low Address, High Address, Image Data>
```

```
...
Send WriteMem Frame <Length, Low Address, High Address, Image Data>
Send Command Frame (Download Cmd<Mode0>)
```

Chapter 6 PCI Host Interface (3842-Based MAC)

6.1 General Overview

The PCI Host Interface allows access to memory and host registers using PCI memory read or write transactions. PRISM firmware supports one double-word transfer through direct access to ISL3874 memory space using Aux port transfers or BAP transfers. These operate in a similar manner to their operation on the HFA3842, which allows quick porting of base-functionality HFA3842 driver code.

Most of the host-side registers have been preserved except where functionality is no longer needed. For example, the attribute FCR registers are not implemented, since attribute space does not exist for the PCI interface. Since only memory space is implemented, PCI I/O read or write operations are not defined.

6.2 PCI Interface Configuration

The PCI core has two sets of configuration registers. One set is read-only and is either configured to default values or set up by ISL3874 firmware on reset. This set is used by the host to determine what type of card this is and what drivers need to be loaded. The other set is the host configuration registers, written by the host to configure various options and responses of the PCI card.

During reset, the core's strapping options cause one of two scenarios to occur for loading the read-only PCI configuration registers:

- If the card is set to power up and run, then ISL3874 firmware is responsible for fetching values from NVRAM and loading them into the proper registers. Note that the interface will be unable to respond to host commands (including configuration commands) until these registers have been loaded.
- If the card is set to power up and go idle, then default values are loaded into the read-only registers so that the PCI interface can be initialized by the host. This mode is most often used when downloading firmware code using the Aux port. Since there is no existing firmware to control the card, the default values allow the host to configure the rest of the interface sufficiently to download code into the memory space of the ISL3874.

The read-only registers set the device id, vendor id, class code, revision id, header type, subsystem id, subsystem vendor id, maximum latency, minimum grant, and the interrupt pin. These registers are all 16-bits wide and are loaded by the DBus destination strobe "i_pciConfigDst". They must be loaded in the following order: {max_lat, min_gnt}, {class_code[23:16], header}, class_code[15:0], {int_pin, rev_id}, subsys_id, subsys_vendor_id, device_id, vendor_id.

The default values are shown in Table 6.1:

Table 6.1: PCI Read-Only Registers

Register	Value	Comment
device_id	0x3873	
vendor_id	0x1260	// Intersil PCI SIG vendor id
class_code	0x02_8000	
subsys_id	0x0000	
subsys_vendor_id	0x1260	// Intersil PCI SIG vendor id
rev_id	0x01	
header	0x00	
max_lat	0x00	
min_gnt	0x00	
int_pin	0x01	// Int A

On reset or power up, several host configuration registers must be written by the host before normal target memory read/write transactions can be used. Target operations are enabled once the Memory Base Address and the Command registers have been written. These are the minimum set of registers required for the card to respond to a target operation.

The Memory Base Address register is used to set the starting address range to which this device will respond. The maximum address space for this chip is 4k. The Command register enables specific features of the PCI host interface. The Memory Access Enable bit must be set to allow any read/write operations. Further information about the PCI configuration registers can be found in the PCI 2.1 Interface Specification (Reference [9]).

6.2.1 RESET

There are two reset pins for this chip. The first, HWResetN, is a hardware reset pin used to reset the entire card on power up. The second reset is the PCI_RST# signal. This is intended to reset only the PCI interface section.

A soft reset is available which does not reset any of the PCI core read-only configuration registers. This soft reset is accomplished in the same manner as with the HFA3842 by writing a one to COR[7]. Note that this register has been moved from its previous location. It now resides at location 0x4C. Only bit 7 (Soft Reset) is implemented for this register. The HCR and COR registers are the only registers that can be written during soft reset. The HCR can be written to override the default MBus strapping options and COR[7] is reset to bring the chip out of soft reset.

LOCK# is not implemented. We do not have atomic accesses and thus have no need to support this. Further, it is not implemented in the mini-PCI specification.

6.2.2 PCI SPECIFIC IMPLEMENTATION

The ISL3874 host side memory space is not intended to be written in a sequential manner, so burst operations are not supported. Only memory read, memory write, and configuration cycles are supported. BAP transfers are supported and should be faster than equivalent PCMCIA BAP transactions. This allows a port of the existing driver from the PCMCIA card to PCI with minimal changes.

Back-to-back operations are not implemented since the trade-off to implement is not supported by the performance gain due to limited MBus bandwidth.

The ISL3874 is a single-function device so only one interrupt, INTA#, is used. An interrupt is generated whenever one of the interrupt sources in the ISR goes active and the corresponding bit in the IMR is enabled. The interrupt pin, INTA#, generates an active low level when requesting an interrupt. HAmask[8] no longer inverts the sense of the interrupt pin as it did on the PCMCIA version.

6.2.3 INTERNAL MEMORY SIZES

Table 6.2 lists the quantity, sizes, and types of internal memory.

Table 6.2: Internal Memory

Quantity	Size	Type
4	1K x 16	SPRAM
2	512 x 8	SPRAM
1	128 x16	DPRAM
1	4K x 16	ROM

6.3 Target Mode Operation

This mode is the default or base mode of communications with the ISL3874 processor. After the host configures the PCI interface itself, PCI memory read and write transactions are used to initialize the processor and to send it commands. These transactions access host-side register addresses in much the same way as the HFA3842 did. Host registers have had their address DWORD aligned (shifted left by one) from the register map used by the HFA3842. This allows ordinary double word accesses to take place on any given host register. Host register addresses are 8-bits wide and wrap at 0xFF in memory space up to the maximum address space. Each register provides up to 16 bits of valid data depending on the PCI read or write system call request length. PCI requests for greater than word length (16-bits) will have the upper bits zeroed.

6.3.1 NORMAL OPERATING MODES

Target mode has three different types of accesses. The biggest difference between them as far as the host is concerned is the amount of time it takes to complete the accesses. The three types are hardware register access, memory-mapped register access and BAP data register access.

Hardware registers complete their access in one M clock cycle, which at normal M clock speeds means the PCI read will complete without a retry. Memory-mapped read cycles will almost always require at least one retry depending on M clock speed and how soon the ISL3874 memory controller grants the memory request. BAP read cycles can fall into either case depending on whether or not a pre-read completes prior to the host requesting another transfer.

The PCI interface supports one level of posted writes. That is, the first write cycle will be accepted and the PCI interface will complete the transaction immediately. If another write occurs before the first write has completed internally, it will not be accepted and the PCI bus will have to retry the write at a later time.

6.4 Functional Differences

Attribute space and I/O space no longer exist. Some of the registers which previously occupied these spaces have been re-mapped to memory space. This includes the HCR and COR registers.

The Configuration and Status registers, the Pin Replacement register and the FCR registers no longer exist.

USB, PCMCIA and CardBus 32 interfaces are not supported.

The wake-up signal now occurs on any recognized PCI transaction to the card’s memory space. The 3842 only issued a wake-up when writing to either the CSR or the EvAck register.

6.4.1 CHANGES FROM THE MINI-PCI

The BBP and MAC processor run from a single 44MHz clock. This clock is brought in on the pin BBP_CLK.

The PCI interface and core run from the 33MHz clock on PCI_CLK.

The supply5v pin used on the mini-PCI bridge chip for ESD protection on 5v tolerant pads is not used since we are using 3.3v PCI pads.

Table 6.3 shows PCI register changes from the HFA3842:

Table 6.3: PCI Register Changes

HAmask	This register contains 16 bits, but most no longer have any special meaning. Only bits 6 and 7, Disable Aux Port and Hardware Buffer Chaining Off respectively, retain their meaning from the HFA3842. The address wrap function occurs after every 256-word block up to the maximum address space of 4k.
FCR	Function Control Registers CSR and PRR no longer exist. They are not required nor supported for PCI operation.
HCR	This register contains 8 write-only bits and is re-mapped to 0x5C in memory space.
COR	Only the Soft Reset bit (7) implements in this register. It moved to memory space at 0x4C.
Master Mode Registers	Several registers have been added that are visible to the host via target mode operations. These are located above 0x7F.

6.5 Memory Map

Table 6.4 show the mapping of PCI memory, organized both by function and by offset address.

Table 6.4: PCI Memory Map

Organized by Function

Memory Offset	Name	Type
00	Command	MM
04	Param0	MM
08	Param1	MM
0C	Param2	MM
10	Status	MM
14	Resp0	MM
18	Resp1	MM
1C	Resp2	MM
20	InfoFID	MM
40	RxFID	MM
44	AllocFID	MM
48	TxComplFID	MM
30	BAP Select0	MM
38	BAP Offset0	MM
6C	BAP Data0	MM
34	BAP Select1	MM
3C	BAP Offset1	MM
70	BAP Data1	MM
60	EvStat (ISR)	HW
64	IntEn (IMR)	HW
68	EvAck	HW
4C	HCR	HW
5C	HCR	HW
28	Control	MM
50	SwSupport0	MM
54	SwSupport1	MM
58	SwSupport2	MM
74	AuxPage	HW
78	AuxOffset	HW

Sorted by offset address

Memory Offset	Name	Type
00	Command	MM
04	Param0	MM
08	Param1	MM
0C	Param2	MM
10	Status	MM
14	Resp0	MM
18	Resp1	MM
1C	Resp2	MM
20	InfoFID	MM
28	Control	MM
30	BAP Select0	MM
34	BAP Select1	MM
38	BAP Offset0	MM
3C	BAP Offset1	MM
40	RxFID	MM
44	AllocFID	MM
48	ComplFID	MM
4C	COR	HW
50	SwSupport0	MM
54	SwSupport1	MM
58	SwSupport2	MM
5C	HCR	HW
60	EvStat (ISR)	HW
64	IntEn (IMR)	HW
68	EvAck	HW
6C	BAP Data0	MM
70	BAP Data1	MM
74	AuxPage	HW
78	AuxOffset	HW
7C	AuxData	(resvd)

Table 6.4: PCI Memory Map

7C	AuxData	(resvd)
80	PCI Master 0 Address High	HW
84	PCI Master 0 Address Low	HW
88	PCI Master 0 Length	HW
8C	PCI Master 0 Control	HW
98	PCI Status	HW
A0	PCI Master 1 Address High	HW
A4	PCI Master 1 Address Low	HW
A8	PCI Master 1 Length	HW
AC	PCI Master 1 Control	HW

80	PCI Master 0 Address High	HW
84	PCI Master 0 Address Low	HW
88	PCI Master 0 Length	HW
8C	PCI Master 0 Control	HW
98	PCI Status	HW
A0	PCI Master 1 Address High	HW
A4	PCI Master 1 Address Low	HW
A8	PCI Master 1 Length	HW
AC	PCI Master 1 Control	HW

Type Legend:
MM = Memory Mapped
HW = Hardware Registers
BA = Buffer Access

Chapter 7 Production Data Area

This chapter describes the Production Data Area, or PDA. The PDA consists of a number of records containing component identity and interface compatibility information. These records are incorporated into firmware images during the download process.

7.1 Definitions

The following definitions are used in this chapter:

Component	A released item; for example, primary firmware, secondary firmware, Miniport driver, NIC.
Component ID	An identification code assigned to identify a particular component.
Interface ID	An Identifier of the interface being described.
NIC	**N**etwork **I**nterface **C**ard.
PDA	**P**roduction **D**ata **A**rea. The PDA is an area programmed in the NIC, which contains a collection of PDRs. The PDA is programmed at manufacturing time.
PDR	**P**roduction **D**ata **R**ecord. A PDR is a record that contains configuration information for a NIC.
Variant	Used to describe a variant of a particular component or interface.

7.2 Identity and Compatibility Information

Components require identity and compatibility information to allow integration of the various components into a larger product set. For example, information must be available to verify compatibility between a particular firmware component and a NIC component.

Components covered in this chapter include:

- NIC boards
- Primary Firmware
- Secondary Firmware
- Tertiary (AP) Firmware
- NDIS Miniport Driver

Components provide version information to identify the release version of the component. Major and minor version release numbers supply version information. This allows release documents to associate bug fixes and product enhancements with a particular release version.

Components also interface with other components, and this interface between components is managed through a compatibility level indication. Interface compatibility is described further in the following sections.

7.2.1 VERSION INFORMATION

Components keep an internal version record that can be retrieved by standard interface mechanisms associated with the component. The general version record format for firmware and NICs is:

Item 1	Component ID
Item 2	Variant
Item 3	Major Version
Item 4	Minor Version

Component ID identifies the component. Currently assigned values are:

- 0x15 = Primary Functions Firmware
- 0x1F = Station Functions Firmware

- 0x8*nnn* = Reserved for NIC IDs.
- 0x14b = Access Point Functions

Variant is a number identifying a variant of a certain component. A variant is typically issued for very minor modifications, such as bug fixes.

Major Version is a number increasing with a new version of the same component with major functionality upgrades.

Minor Version is a number increasing with a new version of the same major version of a component. A minor version typically provides enhancements to existing features.

Version information reported by the NDIS miniport driver only contains the Major and Minor version information.

7.2.2 COMPATIBILITY LEVEL

The components use an actor/supplier *compatibility level* mechanism to keep control of the proper synchronization of the interface between these components. The compatibility level is a numeric value that increases by one each time an element is added to the interface or an existing element is changed. Two terms are defined for the roles of the involved components on each side of the interface:

- *supplier* is the role of the component which “offers” the interface for use
- *actor* is the role of the component which “uses” the interface

By definition, the compatibility level *can only increase* when a new version of the supplier component is released. However, it *need not increase* with each new version.

7.2.2.1 Supplier Compatibility Range

It is possible that a certain version of a component with a supplier role can offer multiple compatibility levels of the interface. This is usually the case when an increase in compatibility level represents an added optional function. If nothing changes in the use of the old set of functions, the previous level will still be offered.

This mechanism allows for the specification of a range of contiguous compatibility levels supplied by a component. The two ends of the range are called *top* and *bottom compatibility level*. The top level represents the full functional capability of this component version. The bottom level is the oldest capability level still offered by this supplier version.

Initial releases of a component with supplier role have top and bottom levels at value 1. The upper compatibility level is incremented when the interface is enhanced or added to without destroying the existing compatibility. The lower compatibility level is incremented when a modification to the interface changes or deletes an existing functionality.

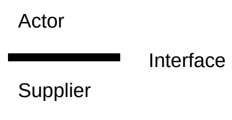
7.2.2.2 Actor Compatibility Range

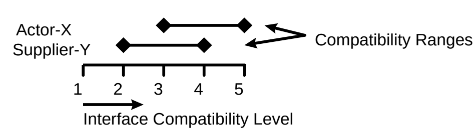
It is possible that a version of a component with an actor role can operate on multiple compatibility levels of the interface. In that case, the component must be able to condition its behavior across the interface depending on the observed top compatibility level of the supplier component.

This mechanism allows for the specification of a range of contiguous compatibility levels supported by a component. The two ends of the range are called *top* and *bottom compatibility level*. The top level represents the latest interface level this component version can fully use. The bottom level is the oldest level with which this component version can still work.

7.2.2.3 Range Overlap

When an actor component and supplier component on the same interface are put together, the requirement for them working together properly is that there is overlap in the two ranges.





7.2.2.4 Compatibility Range Record Format

The basic compatibility range information for firmware and NICs can be retrieved through standard interface mechanisms associated with the component which makes the information available. Two structures are possible.

The first structure indicates support for a single variant of the interface:

Item 1	Role
Item 2	Interface ID
Item 3	Variant Number
Item 4	Bottom Compatibility Level
Item 5	Top Compatibility Level

The second structure indicates support for multiple variants of the interface:

Item 1	Role
Item 2	Interface ID
Item 3	Variant Number *A*
Item 4	Bottom Compatibility Level for variant *A*
Item 5	Top Compatibility Level for variant *A*
Item 3	Variant Number *B*
Item 4	Bottom Compatibility Level for variant *B*
Item 5	Top Compatibility Level for variant *B*
…	…

Item Definitions

Role: The role of the component relative to the interface this record refers to:

- 00 = Supplier
- 01 = Actor

Interface ID: The identifier of the interface this record refers to:

- 1 = Modem-Firmware Interface
- 2 = Controller-Firmware Interface
- 3 = Primary Firmware-Driver Interface
- 4 = Station Firmware-Driver Interface

Variant Number: A number identifying the variant of the interface. When a new variant comes into existence, it is assigned the next number. A new variant implies that the earlier variant will continue to be used and to be maintained and updated. The new variant always starts with a new compatibility level scheme (1:1). To identify preliminary versions of an interface, variant number 0 is used; this should never become a variant number in production versions.

Bottom Compatibility Level: A number representing the lowest compatibility level supported by this interface.

Top Compatibility Level: A number representing the highest compatibility level supported by this interface.

Compatibility information reported by the NDIS miniport driver includes only the Bottom and Top range information.

7.3 PDA File Format

The Production Data Area is a text-format file containing a number of variable-length records, called Production Data Records, or PDRs. Each record consists of at least two data fields separated by a comma, space, tab or carriage return. Comments may be appended to a record, separated from the data fields by a semicolon (;), pound sign (#), or forward-slash (/) character.

Within a PDR, each data field is hexadecimal integer (use of the `0x` prefix to denote hexadecimal notation is optional). The first data field gives the number of following data fields in the record. The second data field gives the PDR code (see Table 7.1). Subsequent fields provide the data for the PDR.

For example, PDR 0x0104 could appear as:

```
2,0x0104,0x07ff    // Channel Set (default for NA)
```

The first data field specifies that there are two following data fields. The second field identifies this record as PDR 0x0104. The third field gives the value specifying the channel set used for North American operation.

Once a PDA has been downloaded to a card, you can erase the value for a PDR by downloading a PDA with a null value for that record. For example, downloading the following PDR:

```
1,0x0104   // Channel Set
```

specifies that there is no channel set defined for the card.

7.4 Production Data Records

The PDA consists of a number of production data records (PDRs). Table 7.1 specifies the current set of PDRs.

Table 7.1: Production Data Records

PDR code	Description
0x0001	Manufacturing part number
0x0002	PDA Version
0x0003	NIC Serial Number
0x0005	NIC RAM Size
0x0006	RF Modem Supplier Range
0x0007	PRISM MAC Supplier Range
0x0008	NIC Identification
0x0101	MAC Address
0x0103	Regulatory Domain List
0x0104	Allowed Channel Set
0x0105	Default Channel
0x0107	Temperature Type
0x0200	IFR Setting
0x0201	RFR Setting
0x0202	Baseline Register Settings
0x0203	Shadow Register Settings
0x0204	IF/RF Register Settings
0x0300	Channel Calibration Set Point
0x0301	Channel Calibration Integrator
0x0400	PRISM II NIC Configuration
0x0401	PRISM USB Identifier
0x0402	PRISM PCI Identifier
0x0403	PRISM PCI Interface Configuration
0x0404	PRISM PCI PM Configuration
0x0405	ZIF Synthesizer Settings
0x0406	RSSI-to-dBm Conversion Constant
0x0407	USB Power Type

Table 7.1: Production Data Records (Continued)

PDR code	Description
0x0409	PRISM USB Max Power
0x0410	USB Manufacture String
0x0411	USB Product String
0x0900	Manufacturing Test Channel Set Points
0x0901	Manufacturing Test Channel Integrators
0x0000	PDA End Record

7.4.1 PDR 0X0001 - MANUFACTURING PART NUMBER

A manufacturing part number is used to uniquely identify the NIC.

PD Record: Manufacturing Part Number		
Field name	**Size (words)**	**Value**
PDR-length	17	17
PDR-code	1	0x0001
Comp ID	16	Part number

7.4.2 PDR 0X0002 - PDA VERSION

The PDA Version record is used to track changes to the PDA and to identify the version level of the PDA installed in the NIC.

PD Record: PDA Version		
Field name	**Size (words)**	**Value**
PDR-length	1	2
PDR-code	1	0x0002
Version	1	Version number

Intersil increments the version level with each change to the PDA.

7.4.3 PDR 0X0003 - NIC SERIAL NUMBER

A NIC Serial Number is used for individual card identification. The serial number format is a 12-character ASCII string and can be applied to a label on the outside of the product. This PDR value can be retrieved through the host API by reading RID 0xFD0A.

PD Record: NIC Serial Number		
Field name	**Size (words)**	**Value**
PDR-length	1	7
PDR-code	1	0x0003
Serial Number	6	

7.4.4 PDR 0X0005 - NIC RAM SIZE

This record identifies the size of the Random Access Memory (RAM) installed on the PRISM card to the firmware. RAM size is specified in “little-endian” format and is in units of Kbytes. If not specified, the firmware image uses a default of 256 Kbytes.

PD Record: NIC RAM Size		
Field name	**Size (words)**	**Value**
PDR-length	1	2
PDR-code	1	0x0005
RAM Size	1	

7.4.5 PDR 0X0006 - RF MODEM SUPPLIER RANGE

This record defines the Supplier Compatibility Range information for the RF Modem component of the NIC. This compatibility range information is used to determine whether certain firmware versions can operate on this version of the RF Modem. The firmware download image contains one or more Actor Compatibility Range information items for the RF Modem. Download utilities will compare this information to determine compatibility.

This record has the following format:

PD Record: RF Modem Supplier Range		
Field name	**Size (words)**	**Value**
PDR-length	1	6
PDR-code	1	0x0006
Role	1	0
Interface ID	1	0x0001
Variant	1	
Bottom Level	1	
Top Level	1	

The variant field identifies the variant of the RF Modem. The value for the PRISM™ I radio is '0' and the value for the PRISM™ II radio is '1'.

NIC Model	Len	PDR	Role	IF ID	Var	Bottom	Top
EVB2-1	5	0006	0	1			
HWB3163	5	0006	0	1	1	1	1
	5	0006	0	1			
	5	0006	0	1			

NOTE: The EVB2-1 (HFA3841EVAL1) does not have an integrated RF Modem. The download utility must ignore this compatibility check when installing new firmware on this device.

This PDR value can be retrieved through the host API by reading RID 0xFD0C.

7.4.6 PDR 0X0007 - PRISM MAC SUPPLIER RANGE

For 3842 platforms, the PRISM MAC supplier Range PDR sets the variant number to 2. Host software can query this record to determine the MAC type installed on the NIC.

PD Record: PRISM MAC Supplier Range		
Field name	**Size (words)**	**Value**
PDR-length	1	6
PDR-code	1	0x0007
Role	1	0
Interface ID	1	
Variant	1	
Bottom Level	1	
Top Level	1	

The variant field is used to identify the variant of the MAC controller, as follows:

MAC ID	Len	PDR	Role	IF ID	Var	Bottom	Top
3841	5	0007	0	2	1	1	1
3842	5	0007	0	2	2	1	1

The value of this PDR can be retrieved through the host API by reading RID 0xFD0D.

7.4.7 PDR 0X0008 - NIC IDENTIFICATION

A NIC Identification record identifies the card type. The format of the record is as follows:

PD Record: NIC Identification		
Field name	**Size (words)**	**Value**
PDR-length	1	5
PDR-code	1	0x0008
Comp ID	1	
Variant	1	
Major Version	1	
Minor Version	1	

Currently allocated values are:

NIC Model	Len	PDR	Comp ID	Var	Major Ver	Minor Ver
EVB2 (HFA3841EVAL1)	5	0008	8000	0	1	0
HWB3763 Rev B	5	0008	8001	0	1	0
HWB3163-01,02,03,04 Rev A	5	0008	8002	0	1	0
HWB3163 Rev B	5	0008	8003	0	1	0
EVB3 (HFA3842EVAL1, Rev B1)	5	0008	8004	0	1	0
HWB1153 PRISM™ I Ref	5	0008	8007	0	1	0

Identified values reserved for customers:

	Len	PDR	Comp ID	Var	Major Ver	Minor Ver
	5	0008	8002	1	1	0
	5	0008	8003	1	1	0
	5	0008	8005	0	1	0
	5	0008	8006	0	1	0

Component IDs for new NICs are allocated starting from 0x8000, with 0xFFFF reserved to indicate a card that does not have this PDR. This PDR value can be retrieved through the host API by reading RID 0xFD0B.

7.4.8 PDR 0X0101 - MAC ADDRESS

This record specifies the unique universal MAC address that is assigned to each NIC. The format for the record is:

PD Record: MAC Address		
Field name	**Size (words)**	**Value**
PDR-length	1	4
PDR-code	1	0x0101
MAC address	3	

The MAC address is specified as three hexadecimal words in reverse byte order. Thus, a value of `0x2000`,`0x00CA`,`0x1913` specifies a MAC address of `00-20-CA-00-13-19`

This PDR value can be retrieved through the host API by reading RID 0xFC01.

7.4.9 PDR 0X0103 - REGULATORY DOMAIN LIST

This record specifies the regulatory domains to which this NIC may be deployed. The regulatory domains are defined in Reference [1].

PD Record: Regulatory Domains		
Field name	**Size (words)**	**Value**
PDR-length	1	7
PDR-code	1	0x0103
Num. Codes	1	
Domain Codes	5	

Up to ten domain codes may be specified. The domain code fields beyond the number given shall be filled with value 0x00.

Domain codes as defined by the 802.11 specification are:

0x10	USA
0x20	Canada
0x30	Most of Europe/Australia
0x31	Spain
0x32	France
0x40	Japan
0x41	Japan (all)

Multiple domains can be specified for a NIC if the regulatory domains have common operating requirements (Channel list restrictions, power control restrictions, etc.).

Logical groupings of compatible regulatory environments are listed below. The firmware does not use the zoning directly, but reports domain codes, allowed channels, and default channels. The following table lists the currently identified zones based on the commonality of the domain regulatory requirements.

Zone 1	North America (NA) = USA and Canada
Zone 2	Europe/CEPT (ETS) = Most of Europe, Australia
Zone 3	France (FR)
Zone 4	Japan (JP)
Zone 5	Spain (SP)

This PDR value can be retrieved through the host API by reading RID 0xFD11.

7.4.10 PDR 0X0104 - ALLOWED CHANNEL SET

This record specifies the allowed channel set for this NIC. This record is directly tied to the regulatory domain specified (see 7.4.9 PDR 0x0103 - Regulatory Domain List).

The format for this record is:

PD Record: Allowed Channel Set		
Field name	**Size (words)**	**Value**
PDR-length	1	2
PDR-code	1	0x0104
Channel Bitmap	1	

The channel bitmap uses a value of 1 to indicate the associated channel is allowed, and has the following format:

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	0	0	Chan 14	Chan 13	Chan 12	Chan 11	Chan 10	Chan 9	Chan 8	Chan 7	Chan 6	Chan 5	Chan 4	Chan 3	Chan 2	Chan 1

The regulatory domain influences the channel set. Firmware does not use zoning directly, but reports the domain codes, allowed channels, and default channels.

Zone	Domain	Channels
Zone 1	NA	channels 1-11
Zone 2	ETS	channels 1-13
Zone 3	FR	channels 10-13
Zone 4	JP	channel 14 {1-14}*
Zone 5	SP	channels 10-11

This PDR value can be retrieved through the host API by reading RID 0xFD10.

7.4.11 PDR 0X0105 - DEFAULT CHANNEL

This record specifies the default channel number to be used by this NIC. This value is directly tied to the regulatory zone specified for this NIC (see 7.4.9 PDR 0x0103 - Regulatory Domain List).

The format for this record is:

PD Record: Default Channel		
Field name	**Size (words)**	**Value**
PDR-length	1	2
PDR-code	1	0x0105
Channel number	1	

The default regulatory domain channel sets are 3, 11, and 14.

This PDR value can be retrieved through the host API by reading RID 0xFC03.

7.4.12 PDR 0X0107 - TEMPERATURE TYPE

This record specifies the operational temperature range of the NIC as defined for the IEEE 802.11 MIB. The format of the record is:

PD Record: Temperature Type		
Field name	**Size (words)**	**Value**
PDR-length	1	2
PDR-code	1	0x0107
Temp Type	1	

Specification of the temperature type uses the following values:

- 0x01 = Type 1, Commercial range 0 ⇔ +40°C
- 0x02 = Type 2, Industrial range –20 ⇔ +55°C
- 0x03 = Type 3, Industrial range –30 ⇔ +70°C

This PDR value can be retrieved through the host API by reading RID 0xFD12.

7.4.13 PDR 0X0200 - IFR SETTING

This record specifies the IF reference setting for the PRISM™ I Intersil radio. This record, in conjunction with the RFR Setting record (see 7.4.14 PDR 0x0201 - RFR Setting), defines the characteristics of the PRISM™ I Intersil Radio on the NIC. Currently, there are only three settings for this record: 22 MHz clock, 44 MHz clock, and 44 MHz clock with PK7 as input. The format for the record is:

PD Record: IFR Setting		
Field name	**Size (words)**	**Value**
PDR-length	1	4
PDR-code	1	0x0200
IFR Setting	3	

Values for IFR Setting are:

Characteristic	Setting
22 MHz	0006,0000,0058
44 MHz	0006,0000,00B0
44 MHz + PK7In	0016,0000,00B0

7.4.14 PDR 0X0201 - RFR SETTING

This record specifies the RF reference setting for the PRISM I Intersil radio. This record, in conjunction with the IFR Setting record (see 7.4.13 PDR 0x0200 - IFR Setting), defines the characteristics of the PRISM™ I Intersil Radio on the NIC. Currently, there are only three settings for this Item: 22 MHz clock, 44 MHz clock, and 44 MHz clock with PK7 as input. The format for the record is:

PD Record: RFR Setting		
Field name	**Size (words)**	**Value**
PDR-length	1	4
PDR-code	1	0x0201
IFR Setting	3	

Values for RFR Setting are:

Characteristic	Setting
22 MHz	0006,0000,005A
44 MHz	0006,0000,00B2
44 MHz + PK7In	0016,0000,00B2

7.4.15 PDR 0X0202 - BASELINE REGISTER SETTINGS

This record specifies the baseline register settings for the Baseband chip. Inclusion of this plug record allows customizable settings of the register set.

The table has 51 word entries in which customized settings can be created. The format for each entry is defined with the following byte order:

Baseband Register Value
Baseband Register Number

The following table gives the format for this record (where *nn* denotes variable data).

PD Record: Baseline Register Settings		
Field name	**Size (words)**	**Value**
PDR-length	1	51 (0x33)
PDR-code	1	0x0202
TX/RX Configure	1	*nn*, 0x12
I/O Polarity	1	*nn*, 0x02
I Cover Code	1	*nn*, 0x04

PD Record: Baseline Register Settings (Continued)		
Field name	**Size (words)**	**Value**
Q Cover Code	1	*nn*, 0x06
TX Preamble Length	1	*nn*, 0x08
TX Rate Selection	1	*nn*, 0x0A
TX Service Field	1	*nn*, 0x0C
TX Length (high)	1	*nn*, 0x0E
TX Length (low)	1	*nn*, 0x10
TX/RX Configure	1	*nn*, 0x12
RX Configure	1	*nn*, 0x14
RX/TX Configure	1	*nn*, 0x16
A/D Test Mode 1	1	*nn*, 0x18
A/D Test Mode 2	1	*nn*, 0x1A
A/D Test Mode 3	1	*nn*, 0x1C
AGC gainClip	1	*nn*, 0x1E
AGC lowerSatCnt	1	*nn*, 0x20
AGC timerCnt	1	*nn*, 0x22
AGC hiSat	1	*nn*, 0x24
AGC lockinLevel	1	*nn*, 0x26
AGC lockWindow	1	*nn*, 0x28
AGC Threshold	1	*nn*, 0x2A
AGC Lookup Control	1	0x00, 0x2C
AGC lookup data	1	0x00, 0x2E
AGC loopGain	1	*nn*, 0x30
AGC RX_IF	1	*nn*, 0x32
AGC TestModes	1	*nn*, 0x34
AGC RX_RfThreshold	1	*nn*, 0x36
AGC low SatAtten	1	*nn*, 0x38
AGC min SigAtten	1	*nn*, 0x3A
Carrier Sense 2	1	*nn*, 0x3C
Manual TxPwr Ctrl	1	*nn*, 0x3E
Test Modes 1	1	*nn*, 0x40
Test Modes 2	1	*nn*, 0x42
Test Bus Address	1	*nn*, 0x44
CMF Coefficient Ctrl	1	*nn*, 0x46
Scrambler Seed, long preamble	1	*nn*, 0x48
Scrambler Seed, short preamble	1	*nn*, 0x4A
ED Threshold	1	*nn*, 0x4C
CMF Gain Threshold	1	*nn*, 0x4E
Reserved 0	1	*nn*, 0x50
Reserved 1	1	*nn*, 0x52
Reserved 2	1	*nn*, 0x54
Reserved 3	1	*nn*, 0x56
Reserved 4	1	*nn*, 0x58

PD Record: Baseline Register Settings (Continued)		
Field name	**Size (words)**	**Value**
Reserved 5	1	*nn*, 0x5A
Reserved 6	1	*nn*, 0x5C
Reserved 7	1	*nn*, 0x5E
Reserved 8	1	*nn*, 0x60
Reserved 9	1	*nn*, 0x62

7.4.16 PDR 0X0203 - SHADOW REGISTER SETTINGS

This record allows definition of a custom set of Shadow register settings. The table has 33 double-word entries (4 bytes) in which customized settings can be created. The format for each entry in the value field is defined with the following byte order:

- AGC Register number
- AGC Lookup Control Register (0x2C)
- AGC register value
- AGC Lookup Data Register (0x2E)

The following table gives the format for this record (where *nn* denotes variable data).

PD Record: Shadow Register Settings		
Field name	**Size (words)**	**Value**
PDR-length	1	65 (0x41)
PDR-code	1	0x0203
AGC Address 00	2	0x00, 0x2C, *nn*, 0x2E
AGC Address 01	2	0x01, 0x2C, *nn*, 0x2E
AGC Address 02	2	0x02, 0x2C, *nn*, 0x2E
AGC Address 03	2	0x03, 0x2C, *nn*, 0x2E
AGC Address 04	2	0x04, 0x2C, *nn*, 0x2E
AGC Address 05	2	0x05, 0x2C, *nn*, 0x2E
AGC Address 06	2	0x06, 0x2C, *nn*, 0x2E
AGC Address 07	2	0x07, 0x2C, *nn*, 0x2E
AGC Address 08	2	0x08, 0x2C, *nn*, 0x2E
AGC Address 09	2	0x09, 0x2C, *nn*, 0x2E
AGC Address 0A	2	0x0A, 0x2C, *nn*, 0x2E
AGC Address 0B	2	0x0B, 0x2C, *nn*, 0x2E
AGC Address 0C	2	0x0C, 0x2C, *nn*, 0x2E
AGC Address 0D	2	0x0D, 0x2C, *nn*, 0x2E
AGC Address 0E	2	0x0E, 0x2C, *nn*, 0x2E
AGC Address 0F	2	0x0F, 0x2C, *nn*, 0x2E
AGC Address 10	2	0x10, 0x2C, *nn*, 0x2E
AGC Address 11	2	0x11, 0x2C, *nn*, 0x2E
AGC Address 12	2	0x12, 0x2C, *nn*, 0x2E
AGC Address 13	2	0x13, 0x2C, *nn*, 0x2E
AGC Address 14	2	0x14, 0x2C, *nn*, 0x2E
AGC Address 15	2	0x15, 0x2C, *nn*, 0x2E
AGC Address 16	2	0x16, 0x2C, *nn*, 0x2E

PD Record: Shadow Register Settings (Continued)		
Field name	**Size (words)**	**Value**
AGC Address 17	2	0x17, 0x2C, *nn*, 0x2E
AGC Address 18	2	0x18, 0x2C, *nn*, 0x2E
AGC Address 19	2	0x19, 0x2C, *nn*, 0x2E
AGC Address 1A	2	0x1A, 0x2C, *nn*, 0x2E
AGC Address 1B	2	0x1B, 0x2C, *nn*, 0x2E
AGC Address 1C	2	0x1C, 0x2C, *nn*, 0x2E
AGC Address 1D	2	0x1D, 0x2C, *nn*, 0x2E
AGC Address 1E	2	0x1E, 0x2C, *nn*, 0x2E
AGC Address 1F	2	0x1F, 0x2C, *nn*, 0x2E

7.4.17 PDR 0X0204 - IF/RF SYNTHESIZER REGISTER SETTINGS

This record specifies the IF/RF Synthesizer Register Settings. Note that each register is represented by two words, but that only the first five nibbles are used to program the registers. For example, 0x1234, 0x5678 would only use 0x1234 and 0x5 with ‘678’ not being used. The format for the record is:

PD Record: IF/RF Synthesizer Register Settings		
Field name	**Size (words)**	**Value**
PDR-length	1	41 (0x)
PDR-code	1	0x0204
IF M Register	2	
IF Rx Cal Register	2	
IF R Counter	2	
IF A/B Counter	2	
RF M Register	2	
RF R Counter	2	
RF A/B Channel 1	2	
RF A/B Channel 2	2	
RF A/B Channel 3	2	
RF A/B Channel 4	2	
RF A/B Channel 5	2	
RF A/B Channel 6	2	
RF A/B Channel 7	2	
RF A/B Channel 8	2	
RF A/B Channel 9	2	
RF A/B Channel 10	2	
RF A/B Channel 11	2	
RF A/B Channel 12	2	
RF A/B Channel 13	2	
RF A/B Channel 14	2	

7.4.18 PDR 0X0300 - CHANNEL CALIBRATION SET POINT

This record specifies the closed-loop power calibration set points for each channel. The format for the record is:

PD Record: Channel Calibration Set Point		
Field name	**Size (words)**	**Value**
PDR-length	1	15 (0x0F)
PDR-code	1	0x0300
Channel 1 Set Point	1	0x0000–0x00FF
Channel 2 Set Point	1	0x0000–0x00FF
Channel 3 Set Point	1	0x0000–0x00FF
Channel 4 Set Point	1	0x0000–0x00FF
Channel 5 Set Point	1	0x0000–0x00FF
Channel 6 Set Point	1	0x0000–0x00FF
Channel 7 Set Point	1	0x0000–0x00FF
Channel 8 Set Point	1	0x0000–0x00FF
Channel 9 Set Point	1	0x0000–0x00FF
Channel 10 Set Point	1	0x0000–0x00FF
Channel 11 Set Point	1	0x0000–0x00FF
Channel 12 Set Point	1	0x0000–0x00FF
Channel 13 Set Point	1	0x0000–0x00FF
Channel 14 Set Point	1	0x0000–0x00FF

7.4.19 PDR 0X0301 - CHANNEL CALIBRATION INTEGRATOR

This record specifies the closed-loop power calibration integrator settings for each channel. The format for the record is:

PD RECORD: Channel Calibration Integrator		
Field name	**Size (words)**	**Value**
PDR-length	1	18 (0x12)
PDR-code	1	0x0301
Channel 1 Integrator	1	0x0000–0x00FF
Channel 2 Integrator	1	0x0000–0x00FF
Channel 3 Integrator	1	0x0000–0x00FF
Channel 4 Integrator	1	0x0000–0x00FF
Channel 5 Integrator	1	0x0000–0x00FF
Channel 6 Integrator	1	0x0000–0x00FF
Channel 7 Integrator	1	0x0000–0x00FF
Channel 8 Integrator	1	0x0000–0x00FF
Channel 9 Integrator	1	0x0000–0x00FF
Channel 10 Integrator	1	0x0000–0x00FF
Channel 11 Integrator	1	0x0000–0x00FF
Channel 12 Integrator	1	0x0000–0x00FF
Channel 13 Integrator	1	0x0000–0x00FF
Channel 14 Integrator	1	0x0000–0x00FF
Channel Backoff 3db	1	0x0000–0x00FF
Channel Backoff 6db	1	0x0000–0x00FF
Channel Backoff 12db	1	0x0000–0x00FF

7.4.20 PDR 0X0400 - 3842 PRISM II NIC CONFIGURATION

The 3842 PRISM II platform is built with additional configurations which require runtime decisions on how to configure the NIC. Configuration items on which the firmware requires information are Radio PE/PD, RAM configuration, PCI configuration, and LED configuration.

This PDR specifies configuration parameters for the PRISM II NIC. The format for the record is:

PD Record: PRISM II NIC Configuration		
Field name	**Size (words)**	**Value**
PDR-length	1	2
PDR-code	1	0x0400
Configuration bitmap	1	

The configuration bitmap has the following format:

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	0	0	0	0	0	0	0	0	0	0	LED CNFG			PCI	RAM	PD/PE

where:

LED CNFG = 000 (Mode1) - Legacy, single LED radio state (default)
001 (Mode 2) - FAA Dual LED
010 (Mode 3) - Dual LED
011 (Mode 4) - Single LED with Data Activity
100 (Mode 5) - reserved
101 (Mode 6) - reserved
110 (Mode 7) - reserved
111 (Mode 8) - reserved

PCI =0 for NIC cards configured for USB or PCMCIA operation
1 for NIC cards configured for PCI operation.

RAM = 0 for NIC cards configured to use 2(x8) RAM
1 for NIC cards configured to use 1(x16) RAM

PD/PE = 0 for NIC cards configured to use Radio PD
1 for NIC cards configured to use Radio PE

LED configuration applies to station operation. Values for bits 3–5 produce the following results:

LED	State	Mode1	Mode2	Mode3	Mode4
LED1	Off	All other states			
	On	Radio on and associated	Radio transmit enabled	Radio on and associated	Radio on and associated
	Blink	Radio on and scanning	N/A	Radio on and scanning	Radio on and scanning
	Flash	N/A	N/A	N/A	Radio on, associated, and TX/RX data
LED2	Off	N/A	Radio transmit disabled	N/a	N/A
	On	N/A	N/A	N/A	N/A
	Flash	N/A	TX/RX data	TX/RX data	N/A

7.4.21 PDR 0X0401 - PRISM USB IDENTIFIER

This PDR specifies the USB Vendor Identification (VID) and USB Product Identification (PID). Manufacturers can replace the Intersil default values with values that correctly identify the components used in their cards.

The format for the record is:

PD Record: PRISM USB Identifier		
Field name	**Size (words)**	**Value**
PDR-length	1	3
PDR-code	1	0x0401
VID	1	
PID	1	

7.4.22 PDR 0X0402 - PRISM PCI IDENTIFIER

This PDR specifies the PCI Subsystem Identification (SID) and Subsystem Vendor Identification (SVID). Manufacturers can replace the Intersil default values with values that correctly identify the components used in their cards.

The format for the record is:

PD Record: PRISM USB Identifier		
Field name	**Size (words)**	**Value**
PDR-length	1	3
PDR-code	1	0x0402
SID	1	
SVID	1	

7.4.23 PDR 0X0403 - PRISM PCI INTERFACE CONFIGURATION

This PDR specifies values for the PCI Interface Configuration register block. PDR 0x0402 allows manufactures to specify only the SID and SVID files; this PDR allows manufacturers to specify values for the entire register block. However, the preferred method to change the register block is through PDR 0x0402.

The format for the record is:

PD Record: PRISM PCI Interface Configuration		
Field name	**Size (words)**	**Value**
PDR-length	1	3
PDR-code	1	0x0403
Maximum Latency [15:8] and Minimum Grant [7:0]	1	
Bits [23:16] of Class Code in [15:8] and Header Type in [7:0]	1	
Class Code bits [15:0]	1	
Interrupt Pin in [15:8] and Revision ID in [7:0]	1	
Subsystem ID (SID)	1	
Subsystem Vendor ID (SVID)	1	
Device ID (DID)	1	
Vendor ID (VID)	1	

7.4.24 PDR 0X0404 - PRISM PM CONFIGURATION

This PDR specifies values for the PCI Power Management register block. Manufacturers can replace the Intersil default values with values that correctly identify the capabilities of their cards.

PCI power management requires firmware support, which at present is not implemented. Therefore, the default value of Status bit 4 is 0 (not supported). When firmware supports power management, the value for Status bit 4 will be 1 with proper values for the Power Management Capabilities PCM register.

The format for the record is:

PD Record: PRISM PCI PM Configuration		
Field name	**Size (words)**	**Value**
PDR-length	1	9
PDR-code	1	0x0404
Status bit 4	1	
Data in [15:8] and PMCSR_BSE in [7:0]	1	
PMCSR	1	
PMC	1	
Next_Ptr [15:8] and Capability ID [7:0]	1	0x0100

7.4.25 PDR 0X0405 - ZIF SYNTHESIZER SETTINGS

This record specifies the ZIF Synthesizer Settings. Note that the register is represented by two words. The format for the record is:

PD Record: ZIF Synthesizer Settings		
Field name	**Size (words)**	**Value**
PDR-length	1	3
PDR-code	1	0x0405
DC Offset Calibration	2	

7.4.26 PDR 0X0406 - RSSI-TO-DBM CONVERSION CONSTANT

This record specifies the value used to convert RSSI values to dBm. The format for the record is:

PD Record: RSSI-to-dBm Conversion Constant		
Field name	**Size (words)**	**Value**
PDR-length	1	2
PDR-code	1	0x0406
dBmAdjust	1	

The value of dBmAdjust is subtracted from the RSSI to approximate dBm values.

This PDR value can be retrieved through the host API by reading RID FC46 (see 4.4.2.1.25 cnfDbmAdjust).

7.4.27 PDR 0X0407 - USB POWER TYPE

This record allows the programmer to override the USB power type specified in the firmware. The format for the record is:

PD Record: USB Power Type		
Field name	**Size (words)**	**Value**
PDR-length	1	2
PDR-code	1	0x0407
USBPowerType	1	

Values can be either 0x8000 (bus powered) or 0xC000 (self powered). The default value is 0x800.

7.4.28 PDR 0X0409 - PRISM USB MAX POWER

This record allows manufacturers to specify the maximum power consumption of a USB device. The format for the record is:

PD Record: PRISM USB Max Power		
Field name	**Size (words)**	**Value**
PDR-length	1	2
PDR-code	1	0x0409
MaxPower	1	09*nn*

This value is returned during USB enumeration and is used by the host to calculate the current power consumption of connected devices.

The value of MaxPower must be in the range of 0900–09FA, where the last two hexadecimal digits of the value specify the maximum current power consumption of the device in 2 ma increments. The default value is 09FA, or 500 ma. The manufacturer must ensure that the value of MaxPower is greater than the current draw of the device.

7.4.29 PDR 0X0410 - USB MANUFACTURE STRING

This record specifies the USB Manufacture string used during enumeration of USB devices. The format for the record is:

PD Record: USB Manufacture String		
Field name	**Size (words)**	**Value**
PDR-length	1	11
PDR-code	1	0x0410
USBManString	10	

The value of USBManString is a the hexadecimal representation of 20 ASCII characters stored in little-endian format. If there are unused characters they must be filled with zeros.

The default ASCII value for this record is “Intersil Americas”.

7.4.30 PDR 0X0411 - USB PRODUCT STRING

This record specifies the USB device string used during USB enumeration. The format for the record is:

PD Record: USB Product String		
Field name	**Size (words)**	**Value**
PDR-length	1	17
PDR-code	1	0x0411
USBProdString	16	

The value of USBProdString is the hexadecimal representation of a string of 32 ASCII characters stored in little-endian format. If there are unused characters they must be filled with zeros.

The default ASCII value for this record is “USB 802.11b WLAN DEVICE”.

7.4.31 PDR 0X0900 - MANUFACTURING TEST CHANNEL SET POINTS

This record specifies the closed-loop power calibration set points established during manufacturing test. This record is used by manufacturing tools and is not used directly by the firmware; it allows a pre-manufacturing step the ability to calibrate the baseband processor to all the different regulatory domains. A final manufacturing step can use this information to create the Channel Calibration Set Point record (see 7.4.18 PDR 0x0300 - Channel Calibration Set Point) used by the firmware.

This record has the following format:

PD Record: Manufacturing Test Channel Set Point		
Field name	**Size (words)**	**Value**
PDR-length	1	31 (0x1F)
PDR-code	1	0x0900
US Filter Channel 1 Set Point	1	0x0000–0x00FF
US Filter Channel 2 Set Point	1	0x0000–0x00FF
US Filter Channel 3 Set Point	1	0x0000–0x00FF
US Filter Channel 4 Set Point	1	0x0000–0x00FF
US Filter Channel 5 Set Point	1	0x0000–0x00FF
US Filter Channel 6 Set Point	1	0x0000–0x00FF
US Filter Channel 7 Set Point	1	0x0000–0x00FF
US Filter Channel 8 Set Point	1	0x0000–0x00FF
US Filter Channel 9 Set Point	1	0x0000–0x00FF
US Filter Channel 10 Set Point	1	0x0000–0x00FF
US Filter Channel 11 Set Point	1	0x0000–0x00FF
US Filter Channel 12 Set Point	1	0x0000–0x00FF
US Filter Channel 13 Set Point	1	0x0000–0x00FF
US Filter Channel 14 Set Point	1	0x0000–0x00FF
ETSI Channel 1 Set Point	1	0x0000–0x00FF
ETSI Channel 13 Set Point	1	0x0000–0x00FF
Japan Channel 1 Set Point	1	0x0000–0x00FF
Japan Channel 2 Set Point	1	0x0000–0x00FF
Japan Channel 3 Set Point	1	0x0000–0x00FF
Japan Channel 4 Set Point	1	0x0000–0x00FF
Japan Channel 5 Set Point	1	0x0000–0x00FF
Japan Channel 6 Set Point	1	0x0000–0x00FF
Japan Channel 7 Set Point	1	0x0000–0x00FF
Japan Channel 8 Set Point	1	0x0000–0x00FF
Japan Channel 9 Set Point	1	0x0000–0x00FF
Japan Channel 10 Set Point	1	0x0000–0x00FF
Japan Channel 11 Set Point	1	0x0000–0x00FF
Japan Channel 12 Set Point	1	0x0000–0x00FF
Japan Channel 13 Set Point	1	0x0000–0x00FF
Japan Channel 14 Set Point	1	0x0000-0x00FF

7.4.32 PDR 0X0901 - MANUFACTURING TEST CHANNEL INTEGRATORS

This record specifies the closed-loop power calibration integrator values established during manufacturing test. This record defines manufacturing tools usage and is not used directly by the firmware; it allows a pre-manufacturing step the ability to calibrate the baseband processor to all the different regulatory domains. A final manufacturing step can use this information to create the Channel Calibration Integrator (PDR 0301) used by the firmware.

This record has the following format:

PD Record: Manufacturing Test Channel Integrators		
Field name	**Size (words)**	**Value**
PDR-length	1	34 (0x22)
PDR-code	1	0x0901
US Filter Channel 1 Integrator	1	0x0000–0x00FF
US Filter Channel 2 Integrator	1	0x0000–0x00FF
US Filter Channel 3 Integrator	1	0x0000–0x00FF
US Filter Channel 4 Integrator	1	0x0000–0x00FF
US Filter Channel 5 Integrator	1	0x0000–0x00FF
US Filter Channel 6 Integrator	1	0x0000–0x00FF
US Filter Channel 7 Integrator	1	0x0000–0x00FF
US Filter Channel 8 Integrator	1	0x0000–0x00FF
US Filter Channel 9 Integrator	1	0x0000–0x00FF
US Filter Channel 10 Integrator	1	0x0000–0x00FF
US Filter Channel 11 Integrator	1	0x0000–0x00FF
US Filter Channel 12 Integrator	1	0x0000–0x00FF
US Filter Channel 13 Integrator	1	0x0000–0x00FF
US Filter Channel 14 Integrator	1	0x0000–0x00FF
ETSI Channel 1 Integrator	1	0x0000–0x00FF
ETSI Channel 13 Integrator	1	0x0000–0x00FF
Japan Channel 1 Integrator	1	0x0000–0x00FF
Japan Channel 2 Integrator	1	0x0000–0x00FF
Japan Channel 3 Integrator	1	0x0000–0x00FF
Japan Channel 4 Integrator	1	0x0000–0x00FF
Japan Channel 5 Integrator	1	0x0000–0x00FF
Japan Channel 6 Integrator	1	0x0000–0x00FF
Japan Channel 7 Integrator	1	0x0000–0x00FF
Japan Channel 8 Integrator	1	0x0000–0x00FF
Japan Channel 9 Integrator	1	0x0000–0x00FF
Japan Channel 10 Integrator	1	0x0000–0x00FF
Japan Channel 11 Integrator	1	0x0000–0x00FF
Japan Channel 12 Integrator	1	0x0000–0x00FF
Japan Channel 13 Integrator	1	0x0000–0x00FF
Japan Channel 14 Integrator	1	0x0000–0x00FF
Channel Backoff 3db	1	0x0000–0x00FF
Channel Backoff 6db	1	0x0000–0x00FF
Channel Backoff 12db	1	0x0000-0x00FF

7.4.33 PDR 0X0000 - PDA END RECORD

This record marks the end of the PDA. It must be the last record in the PDA. It contains a CRC-16 check on the whole PDA. The format for the record is:

PD Record: PDA End Record		
Field name	**Size (words)**	**Value**
PDR-length	1	2
PDR-code	1	0x0000
CRC-16	1	

The CRC-16 calculation starts at the fixed PDA start address and ends (and includes) the PDR code field of this record. When the CRC algorithm is applied to the whole area, including the CRC, the result must be 0.

7.5 Firmware Identity and Compatibility

Primary firmware and secondary firmware are separate components with their own version identifiers and Actor/Supplier compatibility ranges. It is important that releases of these components use valid values for identification and compatibility range RIDs to align configuration management mechanisms with the concepts intended for release versions.

RID FD02 and RID FD20 retrieve the version number and variant information for primary and secondary firmware. The format of RID FD02 and RID FD20 is:

Firmware	Len	RID	Comp ID	Var	Major Ver	Minor Ver
Primary	5	FD02	0015	n	n	n
Secondary	5	FD20	001F	n	n	n

A primary firmware component has a component ID of 0x15. A secondary firmware component has a component ID of 0x1F.

Primary and secondary firmware also have supplier compatibility RIDs to define the compatibility range of the host/firmware interface.

Firmware	Len	RID	Role	IF ID	Var	Bottom	Top
Primary	6	FD03	0	3	n	n	n
Secondary	6	FD21	0	4	n	n	n

Chapter 8 Download Images

8.1 Download Image Format

The download image created by the assembler uses a Motorola S3/S7 hexadecimal load record format.

8.1.1 S3 RECORD FORMAT

The S3 record has the following format:

```
S3##aaaaaaaadd….ddcc<CRLF>
```

where:

- `##` is 2 characters specifying the number of bytes (including address, data and checksum) in the record
- `aa` is 8 characters specifying the address where the data must be stored (represented in big-endian format)
- `dd` is ASCII data bytes
- `cc` is two 2 characters specifying the ASCII bytes one's complement of the checksum of the binary record bytes, including the record length, address, and data

`NOTE:` Note that the S3 records mix the 'endianness' of the data format. The address field is 'big-endian' and the data field is 'little-endian'.

8.1.2 S7 RECORD FORMAT

The S7 record has the following format:

```
S7##aaaaaaaacc<CRLF>
```

where:

- `##` is two characters specifying the number of bytes (including address and checksum) in the record
- `aa` is 8 characters specifying the start address of the code being downloaded (represented in big-endian format)
- `cc` is 2 characters specifying the ASCII bytes one's complement of the checksum of the binary record bytes, including the record length and address

The address field is non-zero on the RAM download images. When performing a RAM download (mode 1 of the download command), the Param0 register receives the low-order 16 bits of this address and the Param1 register receives the high-order 16 bits of this address.

Firmware images targeted to the flash have a zero value for this address.

8.1.3 SPECIAL S3 RECORDS

In addition to containing the "unplugged" load-image of the component to download, the hexadecimal file contains additional special S3 records. These special S3 records, in the address range of 0xFF000000 and above, are reserved for passing configuration and identity information and identifying "plug-information" to user applications. The records include identification of plug records, CRC generation request, and firmware identity records:

Address	Description
`FF000000`	Plug Record
`FF100000`	CRC-16 Generation Record
`FF200000`	Firmware Information Record

8.1.3.1 S3 Plug Records

S3 Plug Records contained in the download image identify areas in the firmware load image to "plug". These records are matched with the PDA records and, if the record is found in both locations, the load image is plugged with the value found in the PDA.

The S3 Plug Records have the following format:

```
S3##FF000000xxxxxxxxaaaaaaaassssssssscc<CRLF>
```

where:

- `##` is 2 characters specifying the total length of the S3 record
- `x` is 8 characters specifying the PDR number (in little-endian format unsigned long value)
- `a` is 8 characters specifying the address location in the load image to plug (in little endian format unsigned long value)
- `s` is 8 characters specifying the length of data to plug (in little-endian format unsigned long value)
- `cc` is 2 characters specifying the one's complement of the checksum of the binary record

There is a special PDR, number 0xFFFFFFFF, which is a location for plugging an ASCIIZ string. This allows the download utility to 'plug' a string into the load image. Currently, the download utility uses this area to plug the filename from which the download image is being generated. This area is not limited to filenames and can be used for any information, as long as it fits in the size specified. This information can be retrieved from the firmware by reading RID FFFF.

8.1.3.2 S3 CRC-16 Generation Record

The S3 CRC-16 Generation Record identifies a segment of the load image over which a CRC-16 is to be generated. The format of the record is:

```
S3##FF100000aaaaaaaassssssssbbbbbbbbcc
```

where:

- `##` is 2 characters specifying the S3 record length
- `a` is 8 characters specifying the address location in the load image over which to start the CRC-16 calculation (in little-endian format unsigned long value)
- `s` is 8 characters specifying the size of the segment over which to calculate the CRC-16 value (in little-endian format unsigned long value)
- `b` is 8 characters specifying a boolean value indicating whether or not to program the CRC-16 value in the flash
- `cc` is 2 characters specifying the one's complement of the checksum of the binary record

If the record indicates that the CRC-16 is to be written to flash, the address to write to is minus 2 from the address specified in the record.

NOTE: At present, CRC-16 support is not implemented. However, the Primary firmware still expects the Secondary firmware to have a 'signature stamp' of 0xC0DE (or 0xDEC0 in little-endian format) at the address location for the CRC-16 value. Utility programs which support download should interpret the S3 CRC-16 Generation record but program 0xC0DE instead of the real CRC value.

8.1.3.3 S3 Firmware Information Records

The S3 Firmware Information Records are informational records that identify version and actor/supplier capability range information of the load image. Information Records are identified with a value of 0xFF200000 in the address field of the S3 record. Each Information Record may span multiple S3 records; utility programs must handle these special cases.

S3 Firmware Information Records have the following format:

```
S3##FF200000ssttdd…ddcc
```

where:

- `##` is 2 characters specifying the S3 record length
- `s` is 2 characters specifying the size, in words, of the information record being defined (in little-endian unsigned short format)
- `t` is 2 characters specifying the type of information record being defined (in little-endian unsigned short format) where:

  `0x0001` – Version Information Type
  `0x0002` – Compatibility Information Type
  `0x0003` – Build Sequence Information Type
  `0x0004` – Platform information Type

- d is the data for the information record (each data item is in little-endian unsigned short format)
- cc is 2 characters specifying the one's complement of the checksum of the binary record

8.1.3.3.1 Version Information Type

The data portion of a version information record has the following format:

Version Information		
Field name	**Size (words)**	**Value**
Component ID	1	
Variant	1	
Major Version	1	
Minor Version	1	

This format is defined in 7.2.1 Version Information.

Utility software can interpret this component ID to determine if the load image is for Primary or Secondary firmware functions.

8.1.3.3.2 Compatibility Information Type

The data portion of a compatibility information record has the following format:

Compatibility Information		
Field name	**Size (words)**	**Value**
Role	1	
Interface ID	1	
Variant Number	1	
Bottom level compatibility	1	
Top level compatibility	1	

This follows the format defined in 7.2.2.4 Compatibility Range Record Format.

NOTE: Multiple compatibility items can be defined in each record.

8.1.3.3.3 Build Sequence Information Type

The data portion of a build sequence information record has the following format

Build Sequence Information		
Field name	**Size (words)**	**Value**
Build Sequence	1	

The build sequence is a free-form word that can be utilized by the firmware developer to extend the version information of the firmware.

8.1.3.3.4 Platform Information Type

The data portion of a platform information record has the following format:

Platform Information		
Field Name	**Size (words)**	**Value**
Platform ID	1	
Variant Number	1	
Major Version	1	
Minor Version	1	

This record further identifies platforms supported by this firmware. A firmware image can support multiple platforms.

8.2 Download Procedure

Download utility programs observe the following procedure when updating the firmware image in a NIC:

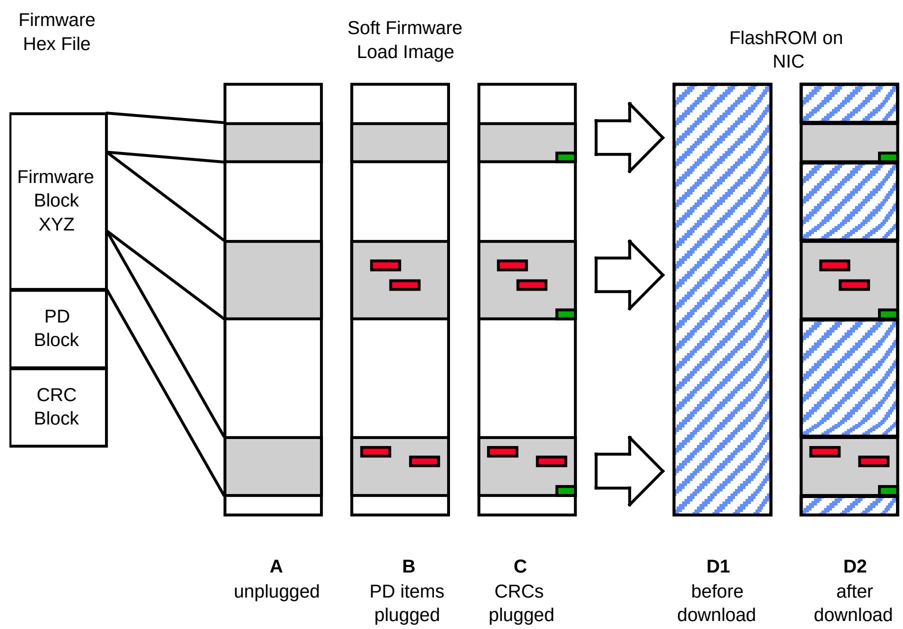


Figure 8.1: Download Procedure

A – The download utility creates an internal representation of the memory map of the Flash ROM, and maps the individual segments of the Firmware block to this map. This is called the soft Firmware Load Image.

B – The download utility interprets the PDA and tries to match all references in the “plug-information” part in the firmware image. For each matched reference, the PDR body is plugged at the appropriate offset in the load image. When a match is not found, either default values are provided or an error is reported (see the following NOTE:).

C – The download uses the CRC Block to calculate CRCs over the plugged segments in the soft image and plugs the CRC values at the intended place in these segments.

D – The download overwrites on the Flash ROM the segments as present in the soft load image, keeping all other parts of the Flash ROM exactly the same as it was before. This is taken care of by the Primary Functions firmware on the NIC.

NOTE: The full body of the PDR is always plugged. To guarantee backwards compatibility (new firmware should run on old NICs), PDRs will never change format and length.

Some records may not have a match in the PDA. In that case, either the default set in the load image is used or the download program can “fake” a value that can be used.

8.2.1 UPDATE UTILITY PROGRAMMER CONSIDERATIONS

The following points must be kept in mind by the programmers of update utilities:

- Not all PDRs are “pluggable” or are plugged
- When all plug record items are written, the plug operation is finished
- When no PDR can be found that matches a certain item in the plug records, the plugging is aborted for that record
- The plug record items give a length in bytes, while the PD Records contain length in words
- If a discrepancy is found in the two lengths, the plug operation is aborted and an error is reported

8.2.2 ADDITIONAL UPDATE CONSIDERATIONS

The download image contains information records that are made available to update utilities. This ensures that the image being downloaded is compatible with the target platform. In addition, the version information in the image can be compared with the installed version on the NIC to determine if this is an upgrade or a downgrade.

When preparing for an image update, the update utility uses the defined RID interface to access the corresponding information in the NIC. The following table shows the relationship between S3 Information records, RIDs, and PDA IDs (when applicable).

S3 Information Type	Name	Related Firmware RID	PDA ID
0001	Firmware image version and type information, where: type 0x15 is primary type 0x1F is secondary	FD02 – Installed primary version FD20 – Installed secondary version	N/A
0002	Compatibility Information between components		
	ID 01 – RF Modem Actor Range	FD0C – RF Modem Supplier Range	0x0006
	ID 02 – Controller Actor Range	FD0D – Controller Supplier Range	0x0007
	ID 03 – Primary Actor Range	FD03 – Primary Supplier Range	N/A
0004	Support NIC Component Platform IDs	FD0B – NIC Component Platform ID	0x0008

The PDA ID is shown for reference only. The PDA ID should only be referenced for “plugging” information by update utilities. The RID interface is the preferred method to access this information.

Appendix A Resource Identifiers

This Appendix provides two tables:

- Table A.1 lists all the Resource IDentifiers (RIDs) supported by the PRISM MAC, sorted numerically by RID number. The final column indicates the section in which the RID is described.
- Table A.2 lists the Information Records provided by the PRISM MAC, sorted numerically by type, and with the section in which the record is described indicated in the final column,

Table A.1: PRISM MAC Resource Identifiers

RID	Name	Description	Section
FC00	cnfPortType	[STA] Connection control characteristics	4.4.2.1.1
FC01	cnfOwnMACAddress	[STA] MAC Address of this node	4.4.2.1.2
FC02	cnfDesiredSSID	[STA] Service Set identification for connection	4.4.2.1.3
FC03	cnfOwnChannel	Communications channel for BSS creation	4.4.2.1.4
FC04	cnfOwnSSID	IBSS creation (STA) or ESS (AP) Service Set Identification	4.4.2.1.5
FC05	cnfOwnATIMWindow	[STA] ATIM Window time for IBSS creation	4.4.2.1.6
FC06	cnfSystemScale	System Scale that specifies the AP density	4.4.2.1.7
FC07	cnfMaxDataLength	Maximum length of MAC Frame Body data	4.4.2.1.8
FC08	cnfWDSAddress	[STA] MAC Address of corresponding WDS Link node	4.4.2.1.9
FC09	cnfPMEnabled	[STA] Switch for ESS Power Management (PM) On/Off	4.4.2.2.1
FC0A	cnfPMEPS	[STA] Switch for ESS PM EPS/PS Mode	4.4.2.2.2
FC0B	cnfMulticastReceive	[STA] Switch for ESS PM Multicast reception On/Off	4.4.2.1.10
FC0C	cnfMaxSleepDuration	[STA] Maximum sleep time for ESS PM	4.4.2.1.11
FC0D	cnfPMHoldoverDuration	[STA] Holdover time for ESS PM	4.4.2.2.3
FC0E	cnfOwnName	Identification text for diagnostic purposes	4.4.2.1.12
FC10	cnfOwnDTIMPeriod	[AP] Beacon intervals between successive DTIMs	4.4.2.1.13
FC11	cnfWDSAddress1	[AP] Port 1 MAC Address of corresponding WDS Link node	4.4.2.1.14
FC12	cnfWDSAddress2	[AP] Port 2 MAC Address of corresponding WDS Link node	4.4.2.1.14
FC13	cnfWDSAddress3	[AP] Port 3 MAC Address of corresponding WDS Link node	4.4.2.1.14
FC14	cnfWDSAddress4	[AP] Port 4 MAC Address of corresponding WDS Link node	4.4.2.1.14
FC15	cnfWDSAddress5	[AP] Port 5 MAC Address of corresponding WDS Link node	4.4.2.1.14
FC16	cnfWDSAddress6	[AP] Port 6 MAC Address of corresponding WDS Link node	4.4.2.1.14
FC17	cnfMulticastPMBuffering	[AP] Switch for PM buffering of multicast messages	4.4.2.1.15
FC23	cnfWEPDefaultKeyID	Default key identifier for WEP	4.4.2.2.4
FC24	cnfDefaultKey0	The first of four default WEP keys	4.4.2.2.5
FC25	cnfDefaultKey1	The second of four default WEP keys	4.4.2.2.5
FC26	cnfDefaultKey2	The third of four default WEP keys	4.4.2.2.5
FC27	cnfDefaultKey3	The fourth of four default WEP keys	4.4.2.2.5
FC28	cnfWEPFlags	Flags for WEP configuration	4.4.2.2.6
FC29	cnfWEPKeyMapping	Table for translating mapped WEP keys	4.4.2.2.7
FC2A	cnfAuthentication	Control of authentication process	4.4.2.2.8
FC2B	cnfMaxAssociatedStations	[AP] Limit maximum number of stations allowed to associate	4.4.2.1.16
FC2D	cnfRoamingMode	[STA] Configure roaming behavior	4.4.2.1.17
FC2E	cnfHostAuthentication	[AP] Allow Host to govern authentication process	4.4.2.1.18
FC30	cnfRcvCrcError	Allow frames received with FCS errors to be passed to the Host	4.4.2.1.19

Table A.1: PRISM MAC Resource Identifiers (Continued)

RID	Name	Description	Section
FC32	cnfAltRetryCnt	Set an alternate retry count for specially marked frames	4.4.2.2.9
FC33	cnfBeaconInterval	[AP] Set interval between beacon frames generated by the AP	4.4.2.2.10
FC34	cnfAPPCFInfo	[AP] Set configuration for AP in PCF mode	4.4.2.1.20
FC35	cnfSTAPCFInfo	[STA] Set configuration for CF-Aware Station	4.4.2.1.21
FC37	cnfPriorityQUsage	Configures priority queues available to TX from CP, CFP, or both in a PC BSS.	4.4.2.2.11
FC40	cnfTIMCtrl	Add TIM Control to control setting and clearing of TIM bits.	4.4.2.1.22
FC42	cnfThirty2Tally	Supports 32-bit accumulated tallies	4.4.2.1.23
FC43	cnfEnhSecurity	[AP] Allows configuration of enhanced security items in the AP.	4.4.2.1.24
FC46	cnfDbmAdjust	Unit of adjustment to convert RSSI value to dBm units	4.4.2.1.25
FC80	GroupAddresses	[STA] Multicast MAC Addresses for message reception	4.4.2.2.12
FC81	createIBSS	[STA] Configure auto port type mode	4.4.2.2.13
FC82	FragmentationThreshold	[STA] Fragment length for unicast message transmission	4.4.2.2.14
FC83	RTSThreshold	[STA] Frame length used for RTS/CTS handshake control	4.4.2.2.15
FC84	TxRateControl	[STA] Data rate control for message transmission	4.4.2.2.16
FC85	PromiscuousMode	[STA] Switch for Promiscuous mode reception On/Off	4.4.2.2.17
FC90	FragmentationThreshold0	[AP] Port 0 fragment length for unicast msg transmission	4.4.2.2.18
FC91	FragmentationThreshold1	[AP] Port 1 fragment length for unicast msg transmission	4.4.2.2.18
FC92	FragmentationThreshold2	[AP] Port 2 fragment length for unicast msg transmission	4.4.2.2.18
FC93	FragmentationThreshold3	[AP] Port 3 fragment length for unicast msg transmission	4.4.2.2.18
FC94	FragmentationThreshold4	[AP] Port 4 fragment length for unicast msg transmission	4.4.2.2.18
FC95	FragmentationThreshold5	[AP] Port 5 fragment length for unicast msg transmission	4.4.2.2.18
FC96	FragmentationThreshold6	[AP] Port 6 fragment length for unicast msg transmission	4.4.2.2.18
FC97	RTSThreshold0	[AP] Port 0 frame length for RTS/CTS handshake control	4.4.2.2.19
FC98	RTSThreshold1	[AP] Port 1 frame length for RTS/CTS handshake control	4.4.2.2.19
FC99	RTSThreshold2	[AP] Port 2 frame length for RTS/CTS handshake control	4.4.2.2.19
FC9A	RTSThreshold3	[AP] Port 3 frame length for RTS/CTS handshake control	4.4.2.2.19
FC9B	RTSThreshold4	[AP] Port 4 frame length for RTS/CTS handshake control	4.4.2.2.19
FC9C	RTSThreshold5	[AP] Port 5 frame length for RTS/CTS handshake control	4.4.2.2.19
FC9D	RTSThreshold6	[AP] Port 6 frame length for RTS/CTS handshake control	4.4.2.2.19
FC9E	TxRateControl0	[AP] Port 0 data rate control for message transmission	4.4.2.2.16
FC9F	TxRateControl1	[AP] Port 1 data rate control for message transmission	4.4.2.2.16
FCA0	TxRateControl2	[AP] Port 2 data rate control for message transmission	4.4.2.2.16
FCA1	TxRateControl3	[AP] Port 3 data rate control for message transmission	4.4.2.2.16
FCA2	TxRateControl4	[AP] Port 4 data rate control for message transmission	4.4.2.2.16
FCA3	TxRateControl5	[AP] Port 5 data rate control for message transmission	4.4.2.2.16
FCA4	TxRateControl6	[AP] Port 6 data rate control for message transmission	4.4.2.2.16
FCB0	cnfShortPreamble	Configure use of short preamble with 802.11b	4.4.2.1.26
FCB1	cnfExcludeLongPreamble	Exclude 802.11b long preamble frames	4.4.2.1.27
FCB3	cnfBasicRates	Define the set of rates reported as basic rates	4.4.2.1.28
FCB4	cnfSupportedRates	Define the set of rates supported by this MAC	4.4.2.1.29
FCB5	cnfFallbackCtrl	Configure rate fallback behavior	4.4.2.2.20

Table A.1: PRISM MAC Resource Identifiers (Continued)

RID	Name	Description	Section
FCB6	WEPKeyDisable	Disable a previously configured WEP key	4.4.2.2.21
FCB7	WEPKeyMapIndex	[AP] Specify mapped WEP key for transmission to station	4.4.2.2.22
FCB8	BroadcastKeyID	[AP] WEP key used for broadcast frames when WEP keymapping is used	4.4.2.2.23
FCB9	EntSecFlag	[AP] Enable Intra-BSS security Filtering	4.4.2.2.24
FCBA	cnfPassiveScanCtrl	[STA] Define time that station listens to a passively-scanned channel	4.4.2.1.30
FCE0	TickTime	[PRI] Auxiliary Timer tick interval	4.4.2.3.1
FCE1	ScanRequest	[STA] Host initiate a scan to return a Sweep Results Table	4.4.2.2.25
FCE2	JoinRequest	[STA] Host initiate joining a specified BSS	4.4.2.2.26
FCE3	authenticateStation	[AP] Host initiate authentication of a specified Station	4.4.2.2.27
FCE4	ChannelInfoRequest	[AP] Request for channel info from last sweep	4.4.2.2.28
FCE5	HostScan	[STA] Perform non-destructive sweep of available BSSIDs	4.4.2.2.29
FD00	MaxLoadTime	[PRI] Maximum response time of the Download command.	4.4.3.1.1
FD01	DownloadBuffer	[PRI] Download buffer location and size.	4.4.3.1.2
FD02	PRIIdentity	[PRI] Primary Functions firmware identification.	4.4.3.1.3
FD03	PRISupRange	[PRI] Primary Functions interface; Supplier compatibility range.	4.4.3.1.4
FD04	CFIActRanges	[PRI] Controller interface; Actor compatibility ranges.	4.4.3.1.5
FD0A	NICSerialNumber	[PRI] Network Interface Card serial number.	4.4.3.1.6
FD0B	NICIdentity	[PRI] Network Interface Card identification.	4.4.3.1.7
FD0C	MFISupRange	[PRI] Modem interface, Supplier compatibility range.	4.4.3.1.8
FD0D	CFISupRange	[PRI] Controller interface; Supplier compatibility range.	4.4.3.1.9
FD10	ChannelList	Allowed communications channels.	4.4.3.1.10
FD11	RegulatoryDomains	List of intended regulatory domains.	4.4.3.1.11
FD12	TempType	Hardware temperature range code.	4.4.3.1.12
FD13	CIS	PC Card Standard Card Information Structure	4.4.3.1.13
FD20	STAIdentity	[STA] Station Functions firmware identification.	4.4.3.1.14
FD21	STASupRange	[STA] Station Functions interface; Supplier compatibility range.	4.4.3.1.15
FD22	MFIActRanges	[STA] Modem interface; Actor compatibility ranges.	4.4.3.1.16
FD23	CFIActRanges	[STA] Controller interface; Actor compatibility ranges.	4.4.3.1.17
FD40	PortStatus	[STA] Actual MAC Port connection control status.	4.4.3.2.1
FD41	CurrentSSID	[STA] Identification of the actually connected SS.	4.4.3.2.2
FD42	CurrentBSSID	[STA] Identification of the actually connected BSS.	4.4.3.2.3
FD43	CommsQuality	[STA] Quality of the Basic Service Set connection.	4.4.3.2.4
FD44	CurrentTxRate	[STA] Actual transmit data rate.	4.4.3.2.5
FD45	CurrentBeaconInterval	Interval time for Beacon transmissions and receptions.	4.4.3.2.6
FD46	CurrentScaleThresholds	Actual System Scale thresholds settings.	4.4.3.2.7
FD47	ProtocolRspTime	Maximum time to await a response to a request message.	4.4.3.2.8
FD48	ShortRetryLimit	Maximum number of transmit attempts for short frames.	4.4.3.2.9
FD49	LongRetryLimit	Maximum number of transmit attempts for long frames.	4.4.3.2.10
FD4A	MaxTransmitLifetime	Maximum transmit frame handling duration.	4.4.3.2.11
FD4B	MaxReceiveLifetime	Maximum received frame handling duration.	4.4.3.2.12
FD4C	CFPollable	[STA] Contention Free pollable capability indication.	4.4.3.2.13

Table A.1: PRISM MAC Resource Identifiers (Continued)

RID	Name	Description	Section
FD4D	AuthenticationAlgorithms	Available Authentication Algorithms indication.	4.4.3.2.14
FD4F	PrivacyOptionImplemented	WEP Option availability indication.	4.4.3.2.15
FD51	dBmCommsQuality	dBm normalized values for CommsQuality	4.4.3.2.16
FD80	CurrentTxRate1	[AP] Actual Port 1 transmit data rate.	4.4.3.2.17
FD81	CurrentTxRate2	[AP] Actual Port 2 transmit data rate.	4.4.3.2.17
FD82	CurrentTxRate3	[AP] Actual Port 3 transmit data rate.	4.4.3.2.17
FD83	CurrentTxRate4	[AP] Actual Port 4 transmit data rate.	4.4.3.2.17
FD84	CurrentTxRate5	[AP] Actual Port 5 transmit data rate.	4.4.3.2.17
FD85	CurrentTxRate6	[AP] Actual Port 6 transmit data rate.	4.4.3.2.17
FD86	OwnMACAddress	[AP] Unique local node MAC Address.	4.4.3.2.18
FD88	ScanResults	Passes last sweep results gathered to the host on demand through the RID mechanism.	4.4.3.2.19
FD89	HostScanResults	Returns results of last Host Scan Request	4.4.3.2.20
FD8A	AuthenicationUsed	Authentication algorithm used in current association	4.4.3.2.21
FDC0	PHYType	Physical layer type indication.	4.4.3.3.1
FDC1	CurrentChannel	Actual frequency channel used for transmission.	4.4.3.3.2
FDC2	CurrentPowerState	Actual power consumption status.	4.4.3.3.3
FDC3	CCAMode	Clear channel assessment mode indication.	4.4.3.3.4
FDC6	SupportedDataRates	Data rates capability information.	4.4.3.3.5
FFFE	BuildSequence	Uniquely identifies created extended version number	4.4.3.1.19
FFFF	FWID	Host assignable identifier for the firmware, typically a file name	4.4.3.1.20

Table A.2: Information Records

Info Type	Name	Description	Section
F100	CommunicationTallies	Communications-related statistical counter values.	4.4.5.3.1
F101	ScanResults	Information from network connection points within range.	4.4.5.2.2
F102	ChannelInfoResults	Bit-map of channels scanned during current sweep (AP Only).	4.4.5.2.3
F103	HostScanResults	Information returned from host-initiated non-destructive scan	4.4.5.2.4
F200	LinkStatus	Station connection change.	4.4.5.4.1
F201	AssociationStatus	[AP] association change related to a specific station.	4.4.5.4.2
F202	AuthenticationRequest	[AP] authentication request from a specific station.	4.4.5.4.3
F203	PowerSaveUserCount	[AP] generated when the number of PS stations changes.	4.4.5.4.4
F204	KeyIdChanged (AP Only)	[AP] generated when key-mapped station changes keys	4.4.5.4.5

Appendix B Example NDIS Miniport Driver API

The PRISM MAC NDIS miniport driver provides a set of proprietary NDIS object identifiers (OIDs) defined as constants of the form `OID_`*string*. These OIDs are used by management applications to query or set information in the driver and by extension to the PRISM MAC network adapter.

NOTE: CW10 is a generic platform naming convention. This nomenclature may change in the future.

This appendix assumes the reader is familiar with the NDIS miniport OID interface as defined for Microsoft Windows 95/98/NT (see Reference [6]).

B.1 PRISM MAC NDIS Miniport Release Package

The release package of the miniport driver includes a set of header files that define the user interface to the PRISM MAC miniport driver. These header files include definitions of the OIDs and supporting structures, plus definitions for interfacing with the RIDs in the PRISM MAC network adapter. The released files are:

- `CW10CMD.H` – HFA3841/HFA3842 host command defines.
- `CW10OID.H` – OID interface function definitions.
- `CW10RID.H` – RID defines and RID structure typedefs.
- `CW10UIF.H` – OID constants and structure definitions.
- `DOT11MIB.H` – 802.11 MIB constant definitions.
- `MACIOCTL.H` – PRISM MAC OID constant definitions.

OIDs that can be queried are:

- `OID_ADAPTER_STATE`
- `OID_CURRENT_BSSID`
- `OID_CW10_GETRID`
- `OID_CW10_GET_PDA`
- `OID_CW10_GET_REGS`
- `OID_CW10_GET_TALLIES`
- `OID_CW10_READ_AUX`
- `OID_FW_STRDESC`
- `OID_FW_VERSION`
- `OID_GET_ALLOWED_CHANNELS`
- `OID_GET_DOT11_MIB`
- `OID_GET_DRIVER_DATE`
- `OID_GET_NETWORK_TYPE`
- `OID_GET_RADIO`
- `OID_GET_SERIAL`
- `OID_GET_TX_RATE`
- `OID_GET_WEP_REQUIRED_MODE`
- `OID_LINK_QUALITY`
- `OID_LINK_SIGNAL_STRENGTH`
- `OID_MACIOCTL_GET_DEBUGFLAGS`
- `OID_SUPPLIER_ADI`

OIDs that can be set are:

- `OID_ADAPTER_SOFT_BOOT`
- `OID_CW10_BLOCK`
- `OID_CW10_CLEAR_TALLIES`
- `OID_CW10_CMD`
- `OID_CW10_DISABLE_AUX`
- `OID_CW10_DOWNLOAD_ENABLE_RAM`

- `OID_CW10_DOWNLOAD_RAM`
- `OID_CW10_DOWNLOAD_ROM`
- `OID_CW10_ENABLE_AUX`
- `OID_CW10_SETRID`
- `OID_CW10_SET_PDA` (will not be implemented)
- `OID_CW10_UNBLOCK`
- `OID_CW10_WRITE_AUX`
- `OID_MACIOCTL_SET_DEBUGFLAGS`
- `OID_SET_DOT11_MIB`
- `OID_SET_NETWORK_TYPE`
- `OID_SET_RADIO`
- `OID_SET_TX_RATE`
- `OID_SET_WEP_REQUIRED_MODE`

B.2 OID Command Interface

There are three components of the OID command interface: Query commands, Set commands, and the supporting structures used by the commands. The following sections describe these three components in detail.

B.2.1 NDIS OID QUERY COMMANDS

Query commands ask the driver for information. In response, the driver returns the requested information, which in some cases the driver must request from the PRISM MAC.

B.2.1.1 OID_ADAPTER_STATE

This command retrieves the adapter state in an unsigned short. Returned values are:

Value	Definition	Description
0	NOT_AVAILABLE	Device state is unavailable
1	CONNECTED	Device is in the connected state
2	AUTHENTICATED	Device is authenticated with an Access Point
3	SYNCED	Device is synchronized with an Access Point
4	SCANNING	Device is scanning for an Access Point
5	DISCONNECTED	Device is disconnected from the wireless network
6	INT_TEST_FAIL	Interrupt test failed
7	SEC_FAIL	Secondary firmware failure, primary functional
8	DISABLED	Radio disabled

B.2.1.2 OID_CURRENT_BSSID

This command retrieves the current BSSID in a six-byte buffer (see 4.4.3.2.3 CurrentBSSID).

B.2.1.3 OID_CW10_GETRID

This command is used to retrieve records maintained in the PRISM MAC. The miniport driver expects the information buffer to contain a `RID_STRUCT` (as defined in `CW10RID.H`). The `rid` subfield must be initialized with the RID value to be retrieved.

```
HOOK_REQUEST    hookRequest;
NDIS_STATUS     nNdisStatus;
RID_STRUCT      rid;
rid.rid = RID_COMMS_QUALITY;
hookRequest.NdisRequest.RequestType = NdisRequestQueryInformation;
hookRequest.NdisRequest.DATA.QUERY_INFORMATION.Oid = OID_CW10_GETRID;
hookRequest.NdisRequest.DATA.QUERY_INFORMATION.InformationBuffer = &rid;
hookRequest.NdisRequest.DATA.QUERY_INFORMATION.InformationBufferLength = size
of(rid);
hookRequest.NdisRequest.DATA.QUERY_INFORMATION.BytesWritten = 0;
```

```
hookRequest.NdisRequest.DATA.QUERY_INFORMATION.BytesNeeded = 0;

nNdisStatus = W32N_MakeNdisRequest(
        g_hDevice,
        &hookRequest,
        &g_OverLapped,
        TRUE
        );

if (nNdisStatus) return(FALSE);
```

B.2.1.4 OID_CW10_GET_PDA

This command retrieves the production data area (PDA) from the PRISM MAC network adapter. The input buffer to this query must be at least 1024 bytes to accommodate the PDA.

B.2.1.5 OID_CW10_GET_REGS

This command returns a `CW10_REGISTER_STRUCT` containing the current values of the PRISM MAC Host Interface registers (see 4.1 Interface Registers).

B.2.1.6 OID_CW10_GET_TALLIES

This command returns a `LOCAL_TALLY_STRUCT` containing communications tallies as reported by the PRISM MAC network adapter to the driver (see 4.4.5.3.1 CommunicationTallies).

B.2.1.7 OID_CW10_READ_AUX

This command reads memory on the PRISM MAC through the auxiliary port (AUX). A `CW10_IMAGE_BLOCK_STRUCT` is provided to the driver to handle the request. The address and count fields of the structure are initialized with the required values before the call. If there are no errors, the buffer field contains the data read from the AUX port.

Access to the AUX port is protected by the PRISM MAC and must be explicitly enabled with a set request of `OID_CW10_ENABLE_AUX` (see 4.1.2.5.1 Control Register (I/O offset 14)).

B.2.1.8 OID_FW_STRDESC

This command retrieves the descriptive name of the station firmware residing on the adapter. This is the name of the file from which the secondary firmware image was loaded (for example, `SF010203.hex`).

B.2.1.9 OID_FW_VERSION

This command retrieves the firmware version in an unsigned long (see 4.4.3.1.7 NICIdentity).

The version reported applies to station firmware. The 8-bit values for major revision, minor revision, and variant number are combined as follows: result = (major << 16) + (minor < 8) + variant.

B.2.1.10 OID_GET_ALLOWED_CHANNELS

This command retrieves the allowed channels in an unsigned short. See 4.4.3.1.10 ChannelList for a description of the bitmap layout.

B.2.1.11 OID_GET_DOT11_MIB

This command retrieves the management information base parameters. A `_MIB_Interface_Struct` (as defined in `DOT11MIB.H`) is provided to the driver to handle the request. The `MIB_Id` field is initialized before the call. If there are no errors, the `Length` and `Data` fields contain information from the MIB. The input buffer to this query must be at least 408 bytes to accommodate the longest returned `Data` value.

B.2.1.12 OID_GET_DRIVER_DATE

This command retrieves the driver build data in a null terminated character string that should be 12 bytes long.

B.2.1.13 OID_GET_NETWORK_TYPE

This command retrieves the network type in an unsigned char with one of the following values:

- 0 = NT_IBSS
- 1 = NT_BSS
- 3 = NT_PSIBSS

B.2.1.14 OID_GET_RADIO

This command retrieves the radio status in an unsigned long, where

- 1 = Enabled
- 0 = Disabled

B.2.1.15 OID_GET_SERIAL

This command retrieves the serial number in a 12-byte null-terminated string (see 4.4.3.1.6 NICSerialNumber).

B.2.1.16 OID_GET_TX_RATE

This command retrieves the transmit data rate (in Mbit/s) in an unsigned short. This value is only accurate if the MAC port is in the Enabled state (see 4.4.3.2.5 CurrentTxRate).

B.2.1.17 OID_GET_WEP_REQUIRED_MODE

This command indicates whether Wired Equivalent Privacy (WEP) is enabled (see 4.4.2.2.6 cnfWEPFlags), and if so, the length of WEP keys used. Returned values are:

- 0 = WEP disabled
- 2 = 64-bit WEP enabled
- 6 = 128-bit WEP enabled

B.2.1.18 OID_LINK_QUALITY

This command retrieves link quality in an unsigned char (see 4.4.3.2.4 CommsQuality) with a value in the range of 0–15.

B.2.1.19 OID_LINK_SIGNAL_STRENGTH

This command retrieves signal strength in an unsigned char (see 4.4.3.2.4 CommsQuality) with a value in the range of 0–15.

B.2.1.20 OID_MACIOCTL_GET_DEBUGFLAGS

This command retrieves the debug flags in an unsigned long. The bits in this flag are defined as follows (see the file `TpiDebug.h`):

```
/* Application specific filtering flags */
# define DBG_GEN_SYS         0x00000001L /* Generic subsystem */
# define DBG_HW_SYS          0x00000002L /* Hardware subsystem */
# define DBG_ISR_SYS         0x00000004L /* Interrupt handler */
# define DBG_IOCTL_SYS       0x00000008L /* Ioctl interface */
# define DBG_TX_SYS          0x00000010L /* Transmit handler */
# define DBG_RX_SYS          0x00000020L /* Receive handler */
# define DBG_CALL_SYS        0x00000040L /* Call management */
# define DBG_INIT_SYS        0x00000080L /* Initialization handler */
# define DBG_HALT_SYS        0x00000100L /* Termination handler */
# define DBG_RESET_SYS       0x00000200L /* Reset handler */
# define DBG_TRAP_SYS        0x00000400L /* Trap/exception handler */
# define DBG_REG_SYS         0x00000800L /* Registry interfaces */
# define DBG_ALL_SYS         0xFFFFFFFFL /* Enable all sub-systems */
```

B.2.1.21 OID_SUPPLIER_ADI

This command returns the supplier capability range of the application driver interface. Application programs use this range to determine if the interface implemented in the driver is compatible with what the application expects; see 7.2 Identity and Compatibility Information.

The driver returns a `CW10_SUPPLIER_ADI_STRUCT` with the top and bottom value ranges filled in. It is the responsibility of the application program to verify that the supplier range returned from the driver overlaps with the actor range with which the application was compiled.

The following code snippet from `CW10UIF.H` contains the relevant typedefs and defines used to support this OID.

```
typedef struct _CW10_SUPPLIER_ADI_STRUCT {
   USHORT bottom;
   USHORT top;
} CW10_SUPPLIER_ADI_STRUCT, *PCW10_SUPPLIER_ADI_STRUCT;
//
// Compatibility defines
//
#define DRI_SUPPLIER_ADI_TOP     1
#define DRI_SUPPLIER_ADI_BOTTOM  1

#define DRI_ACTOR_ADI_TOP        DRI_SUPPLIER_ADI_TOP
#define DRI_ACTOR_ADI_BOTTOM     DRI_SUPPLIER_ADI_BOTTOM
```

An application program uses the `DRI_ACTOR_`*string* defines for comparison with data returned from the driver. The driver and the application are both compiled with `CW10UIF.H`. As features are added to the driver, the `DRI_SUPPLIER_ADI_TOP` value is incremented. Older applications can still communicate with the driver if the bottom value has not changed. If the current OIDs are modified, then both `DRI_SUPPLIER_ADI_TOP` and `DRI_SUPPLIER_ADI_BOTTOM` must be modified so that older applications will fail when comparing the capability ranges.

The current supplier range values for the driver ADI is (1,1).

B.2.2 NDIS OID SET COMMANDS

Set commands change the operation of the driver based on the value of the command. In some cases, these commands will cause the driver to issue commands to the PRISM MAC in order to implement the command.

B.2.2.1 OID_ADAPTER_SOFT_BOOT

This command issues a soft boot.

B.2.2.2 OID_CW10_BLOCK

This command causes the driver to block all transmits from higher layer protocols to the PRISM MAC. The transmit requests are returned to the sending protocols with a status of success. This command is needed to prevent accesses to the PRISM MAC when the PRISM MAC is in a state that would cause the driver to report errors on transmits. NDIS will eventually send a reset command to the driver. This will cause problems on certain action sequences such as downloading of new firmware.

B.2.2.3 OID_CW10_CLEAR_TALLIES

This command clears the tallies being maintained by the driver (see 4.4.5.2.1 CommunicationTallies).

B.2.2.4 OID_CW10_CMD

This command is used to pass PRISM MAC host commands through the driver to the PRISM MAC. An initialized `CW10_CMD_STRUCT` is passed to the driver with the commands and parameters needed. On return, the `CW10_CMD_STRUCT` contains the command status and response values.

NOTE: Sending commands directly to the PRISM MAC may have adverse affects on its operation and may further affect the NDIS protocol stacks.

B.2.2.5 OID_CW10_DISABLE_AUX

This command disables read and write access to the AUX port (see 4.1.2.5.1 Control Register (I/O offset 14)).

B.2.2.6 OID_CW10_DOWNLOAD_ENABLE_RAM

This command sets the starting address for a RAM download image sequence. This command is issued first for a RAM image download, followed by several `OID_CW10_DOWNLOAD_RAM` calls to download the image, then an Initialize command to force the PRISM MAC to start running the RAM image.

B.2.2.7 OID_CW10_DOWNLOAD_RAM

This command writes a block of data to volatile RAM. A `CW10_IMAGE_BLOCK_STRUCT` is initialized with the data, the count of the size of data, and the address location to which to write the data. A previous call of `OID_CW10_DOWNLOAD_ENABLE_RAM` is required to set the starting address for the RAM image download.

B.2.2.8 OID_CW10_DOWNLOAD_ROM

This command writes a block of data to NVRAM. A `CW10_IMAGE_BLOCK_STRUCT` is initialized with the data, the count of the size of data, and the address location to which to write the data. The command handles enabling of NVRAM download programming, writing the data to the download buffer in the PRISM MAC, and flushing the data to the NVRAM.

B.2.2.9 OID_CW10_ENABLE_AUX

This command enables read and write access to the AUX port (see 4.1.2.5.1 Control Register (I/O offset 14)).

B.2.2.10 OID_CW10_SETRID

This command is used to set records maintained in the PRISM MAC. The miniport driver expects the information buffer to contain a valid `RID_STRUCT` (as defined in `CW10RID.H`) with all subfields properly initialized to the values desired.

B.2.2.11 OID_CW10_SET_PDA

will not be implemented

B.2.2.12 OID_CW10_UNBLOCK

This command reverses the action of the `OID_CW10_BLOCK` command.

B.2.2.13 OID_CW10_WRITE_AUX

This command writes data via the AUX port to a specified address in the PRISM MAC address space. A `CW10_IMAGE_BLOCK_STRUCT` is initialized with the address, count, and buffer data to be sent to the card.

Access to the AUX port is protected by the PRISM MAC and must be explicitly enabled with a set request of `OID_CW10_ENABLE_AUX`.

B.2.2.14 OID_MACIOCTL_SET_DEBUGFLAGS

This command sets the debug flags from an unsigned long. The bits in this flag are defined as follows (see `TpiDebug.h`):

```
/* Application specific filtering flags */
# define DBG_GEN_SYS         0x00000001L /* Generic subsystem */
# define DBG_HW_SYS          0x00000002L /* Hardware subsystem */
# define DBG_ISR_SYS         0x00000004L /* Interrupt handler */
# define DBG_IOCTL_SYS       0x00000008L /* Ioctl interface */
# define DBG_TX_SYS          0x00000010L /* Transmit handler */
# define DBG_RX_SYS          0x00000020L /* Receive handler */
# define DBG_CALL_SYS        0x00000040L /* Call management */
# define DBG_INIT_SYS        0x00000080L /* Initialization handler */
# define DBG_HALT_SYS        0x00000100L /* Termination handler */
# define DBG_RESET_SYS       0x00000200L /* Reset handler */
# define DBG_TRAP_SYS        0x00000400L /* Trap/exception handler */
# define DBG_REG_SYS         0x00000800L /* Registry interfaces */
# define DBG_ALL_SYS         0xFFFFFFFFL /* Enable all sub-systems */
```

NOTE: Unlike other set commands, `OID_MACIOCTL_SET_DEBUGFLAGS` must use the query function `CW10QueryInformation()` (not `CW10SetInformation()`).

B.2.2.15 OID_SET_DOT11_MIB

This command sets the management information base parameters. A `_MIB_Interface_Struct` (as defined in `DOT11MIB.H`) is provided to the driver to handle the request. The `MIB_Id`, `Length`, and `Data` fields are initialized before the call. If there are no errors, the `Length` and `Data` field as needed.

B.2.2.16 OID_SET_NETWORK_TYPE

This command sets the network type from an unsigned char using one of the following values:

- 0 = NT_IBSS
- 1 = NT_BSS
- 3 = NT_PSIBSS

B.2.2.17 OID_SET_RADIO

This command sets the radio status in an unsigned long, as follows:

- 1 = Enabled
- 0 = Disabled

B.2.2.18 OID_SET_TX_RATE

This command sets the transmit data rate (in Mbit/s) in an unsigned short (see 4.4.2.2.16 TxRateControl).

B.2.2.19 OID_SET_WEP_REQUIRED_MODE

This command enables or disables Wired Equivalent Privacy (WEP) for the PRISM adapter (see 4.4.2.2.6 cnfWEPFlags) using an unsigned char with one of the following values:

- 0 = WEP disabled
- 2 = 64-bit WEP
- 6 = 128-bit WEP

B.2.3 SUPPORTING STRUCTURES

Many OID calls receive parameters or return results through supporting structures. This section lists the definitions for each structure used and specifies the OIDs that use them.

B.2.3.1 CW10_CMD_STRUCT

This structure is used by `OID_CW10_CMD` to send PRISM MAC host interface commands directly to the PRISM MAC. Command responses are returned back by the same structure.

```
typedef struct _CW10_CMD_STRUCT {
    union {
        struct {
            USHORT  cmd;
            USHORT  param0;
            USHORT  param1;
            USHORT  param2;
        } send;
        struct {
            USHORT  status;
            USHORT  resp0;
            USHORT  resp1;
            USHORT  resp2;
        } rsp;
    } u;
} CW10_CMD_STRUCT, *PCW10_CMD_STRUCT;
```

B.2.3.2 CW10_REGISTER_STRUCT

This structure is used by `OID_CW10_GET_REGS` to retrieve the current set of PRISM MAC register values.

```
typedef struct _CW10_REGISTER_STRUCT {
    USHORT  cmd;
    USHORT  param0;
    USHORT  param1;
    USHORT  param2;
    USHORT  status;
    USHORT  resp0;
    USHORT  resp1;
    USHORT  resp2;
    USHORT  infoFID;
    USHORT  rxFID;
    USHORT  allocFID;
    USHORT  txComplFID;

    USHORT  select0;
    USHORT  offset0;
    USHORT  data0;

    USHORT  select1;
    USHORT  offset1;
    USHORT  data1;

    USHORT  evStat;
    USHORT  intEn;
    USHORT  evAck;

    USHORT  control;
    USHORT  auxPage;
    USHORT  auxOffset;

    USHORT  sw0;
    USHORT  sw1;
    USHORT  sw2;
} CW10_REGISTER_STRUCT, *PCW10_REGISTER_STRUCT;
```

B.2.3.3 CW10_IMAGE_BLOCK_STRUCT

This structure is used by the following OIDs:

- `OID_CW10_READ_AUX`
- `OID_CW10_WRITE_AUX`
- `OID_CW10_DOWNLOAD_RAM`
- `OID_CW10_DOWNLOAD_ROM`

```
#define MAX_IMAGE_SIZE  8192

typedef struct _CW10_IMAGE_BLOCK_STRUCT
    ULONG   address;
    USHORT  count;
    UCHAR   buffer[MAX_IMAGE_SIZE];
} CW10_IMAGE_BLOCK_STRUCT, *PCW10_IMAGE_BLOCK_STRUCT;
```

B.2.3.4 CW10_SUPPLIER_ADI_STRUCT

This structure is returned from the driver by an `OID_CW10_SUPPLIER_ADI` query. The top and bottom range number must be compared with the actor values with which the application was compiled. This verifies that the application/driver interface is compatible.

```
//
// Structure used for query of OID_CW10_SUPPLIER_ADI
//
typedef struct _CW10_SUPPLIER_ADI_STRUCT {
    USHORT  bottom;
    USHORT  top;
} CW10_SUPPLIER_ADI_STRUCT, *PCW10_SUPPLIER_ADI_STRUCT;

//
// Compatibility defines
//
// Driver/Application interface compatibility ranges
//  This header file is provided to support application programs that need
//  to interface with the driver and the card. A query of
//  OID_CW10_SUPPLIER_ADI will return the ADI supplier range which the
//  application program needs to compare with the following defines to
//  verify compatibility of the interface (they need to overlap).
//
// Application programs will use the ACTOR defines while the driver will
// use the SUPPLIER defines.
//
#define DRI_SUPPLIER_ADI_TOP    1
#define DRI_SUPPLIER_ADI_BOTTOM 1

#define DRI_ACTOR_ADI_TOP       DRI_SUPPLIER_ADI_TOP
#define DRI_ACTOR_ADI_BOTTOM    DRI_SUPPLIER_ADI_BOTTOM
```

B.2.3.5 CW10_LOCAL_TALLY_STRUCT

This structure defines the communications tallies maintained by the PRISM MAC (see 4.4.5.2.1 CommunicationTallies).

```
//
// Communication Tallies Internal struct
//
typedef struct _LOCAL_TALLY_STRUCT {
    ULONG   txUnicast;
    ULONG   txMulticast;
    ULONG   txFrags;
    ULONG   txUniOctets;
    ULONG   txMultiOctets;
    ULONG   txDeferredTx;
    ULONG   txSingleRetries;
    ULONG   txMultipleRetries;
    ULONG   txRetryLimitExc;
    ULONG   txDiscards;

    ULONG   rxUnicast;
    ULONG   rxMulticast;
    ULONG   rxFrags;
    ULONG   rxUniOctets;
    ULONG   rxMultiOctets;
    ULONG   rxFCSErrors;
    ULONG   rxNoBuffer;
    ULONG   txWrongSA;
    ULONG   rxWEPUndecryp;
    ULONG   rxMsgInMsg;
    ULONG   rxMsgInBadMsg;
    ULONG   huntCmd;
```

```
    ULONG   rxOvr;
    ULONG   retryMany;
    ULONG   retrySingle;
    ULONG   retryExhaust;
    ULONG   crcError;
    ULONG   plcpCrc;
    ULONG   frameTooLong;
    ULONG   retryLong;
    ULONG   retryShort;
    ULONG   expReport;

//AS: April 20, The following are needed for dot11 MIB support
//  they are currently not supported by the firmware, but need to be added.

    ULONG   rxDuplicates;
    ULONG   txRtsSuccess;
    ULONG   txRtsFailure;
    ULONG   txAckFailure;

//  The following is padding to allow more elements to be added without
//  changing the size of the struture in the interface
    ULONG   reserved[8];

} LOCAL_TALLY_STRUCT, *PLOCAL_TALLY_STRUCT;
```

B.2.3.6 _MIB_Interface_Struct

This structure is used by `OID_GET_DOT11_MIB` and `OID_SET_DOT11_MIB` to retrieve and set management information base parameters.

```
typedef struct _MIB_Interface_Struct {
    ULONG   MIB_Id;
    ULONG   Status;
    ULONG   Length;
    UCHAR   Data[1];
} MIB_Interface_Type;
```

The `MIB_Id` should be selected from the defines listed below (see also `DOT11MIB.H`). The defines also indicate the size in bytes of each parameter, and will be copied into the `Length` field of `_MIB_Interface_Struct` by the query or set function. ASN ID cross-references Ids used in the 802.11 Specifications ASN model.

```
//
//      MIB NAME                             MIB ID      // ASN ID       Size
//
#define DOT11STATIONID                       0X01010101  // 1.1.1.1      6
#define DOT11MEDIUMOCCUPANCYLIMIT            0X01010102  // 1.1.1.2      2
#define DOT11CFPOLLABLE                      0X01010103  // 1.1.1.3      1
#define DOT11CFPPERIOD                       0X01010104  // 1.1.1.4      1
#define DOT11CFPMAXDURATION                  0X01010105  // 1.1.1.5      2
#define DOT11AUTHENTICATIONRESPONSETIMEOUT   0X01010106  // 1.1.1.6      4
#define DOT11PRIVACYOPTIONIMPLEMENTED        0X01010107  // 1.1.1.7      1
#define DOT11POWERMANAGEMENTMODE             0X01010108  // 1.1.1.8      1
#define DOT11DESIREDSSID                     0X01010109  // 1.1.1.9      33
#define DOT11DESIREDBSSTYPE                  0X0101010A  // 1.1.1.10     1
#define DOT11OPERATIONALRATESET              0X0101010B  // 1.1.1.11     9
#define DOT11BEACONPERIOD                    0X0101010C  // 1.1.1.12     2
#define DOT11DTIMPERIOD                      0X0101010D  // 1.1.1.13     1
#define DOT11ASSOCIATIONRESPONSETIMEOUT      0X0101010E  // 1.1.1.14     4
```

```
#define DOT11AUTHENTICATIONALGORITHMSINDEX  0X01020101  // 1.2.1.1      1
#define DOT11AUTHENTICATIONALGORITHM        0X01020102  // 1.2.1.2      1
#define DOT11AUTHENTICATIONALGORITHMSENABLE 0X01020103  // 1.2.1.3      1

#define DOT11PRIVACYINVOKED                 0X01050101  // 1.5.1.1      1
#define DOT11WEPDEFAULTKEYID                0X01050102  // 1.5.1.2      1
#define DOT11WEPKEYMAPPINGLENGTH            0X01050103  // 1.5.1.3      1
#define DOT11EXCLUDEUNENCRYPTED             0X01050104  // 1.5.1.4      1
#define DOT11WEPICVERRORCOUNT               0X01050105  // 1.5.1.5      4
#define DOT11WEPEXCLUDEDCOUNT               0X01050106  // 1.5.1.6      4

#define DOT11WEPDEFAULTKEYSTABLE            0X01030000  // 1.3          25
#define DOT11WEPDEFAULTKEYINDEX             0X01030101  // 1.3.1.1      1
#define DOT11WEPDEFAULTKEYVALUE             0X01030102  // 1.3.1.2      5

#define DOT11WEPKEYMAPPINGINDEX             0X01040101  // 1.4.1.1      1
#define DOT11WEPKEYMAPPINGADDRESS           0X01040102  // 1.4.1.2      6
#define DOT11WEPKEYMAPPINGWEPON             0X01040103  // 1.4.1.3      1
#define DOT11WEPKEYMAPPINGVALUE             0X01040104  // 1.4.1.4      5
#define DOT11WEPKEYMAPPINGSTATUS            0X01040105  // 1.4.1.5      1

#define DOT11RTSTHRESHOLD                   0X02010102  // 2.1.1.2      2
#define DOT11SHORTRETRYLIMIT                0X02010103  // 2.1.1.3      1
#define DOT11LONGRETRYLIMIT                 0X02010104  // 2.1.1.4      1
#define DOT11FRAGMENTATIONTHRESHOLD         0X02010105  // 2.1.1.5      2
#define DOT11MAXTRANSMITMSDULIFETIME        0X02010106  // 2.1.1.6      4
#define DOT11MAXRECEIVELIFETIME             0X02010107  // 2.1.1.7      4
#define DOT11MANUFACTURERID                 0X02010108  // 2.1.1.8      129
#define DOT11PRODUCTID                      0X02010109  // 2.1.1.9      129

#define DOT11GROUPADDRESSESINDEX            0X02030101  // 2.3.1.1      1
#define DOT11ADDRESS                        0X02030102  // 2.3.1.2      6
#define DOT11GROUPADDRESSESSTATUS           0X02030103  // 2.3.1.3      1

#define DOT11TRANSMITTEDFRAGMENTCOUNT       0X02020101  // 2.2.1.1      4
#define DOT11MULTICASTTRANSMITTEDFRAMECOUNT 0X02020102  // 2.2.1.2      4
#define DOT11FAILEDCOUNT                    0X02020103  // 2.2.1.3      4
#define DOT11RETRYCOUNT                     0X02020104  // 2.2.1.4      4
#define DOT11MULTIPLERETRYCOUNT             0X02020105  // 2.2.1.5      4
#define DOT11FRAMEDUPLICATECOUNT            0X02020106  // 2.2.1.6      4
#define DOT11RTSSUCCESSCOUNT                0X02020107  // 2.2.1.7      4
#define DOT11RTSFAILURECOUNT                0X02020108  // 2.2.1.8      4
#define DOT11ACKFAILURECOUNT                0X02020109  // 2.2.1.9      4
#define DOT11RECEIVEDFRAGMENTCOUNT          0X0202010A  // 2.2.1.10     4
#define DOT11MULTICASTRECEIVEDFRAMECOUNT    0X0202010B  // 2.2.1.11     4
#define DOT11FCSERRORCOUNT                  0X0202010C  // 2.2.1.12     4
#define DOT11TRANSMITTEDFRAMECOUNT          0X0202010D  // 2.2.1.13     4
#define DOT11WEPUNDECRYPTABLECOUNT          0X0202010E  // 2.2.1.14     4
#define TRANSMITUNICASTOCTETS               0X02020110  // not 802.11   4
#define TRANSMITMULTICASTOCTETS             0X02020111  // not 802.11   4
#define RECEIVEUNICASTOCTETS                0X02020112  // not 802.11   4
#define RECEIVEMULTICASTOCTETS              0X02020113  // not 802.11   4

#define DOT11RESOURCETYPEIDNAME             0X03010100  // 3.1.1        4

#define DOT11RESOURCEINFOTABLE              0X03010200  // 3.1.2        390

#define DOT11CURRENTCHANNEL                 0X04050101  // 4.5.1.1      1
#define DOT11CCAMODESUPPORTED               0X04050102  // 4.5.1.2      1
#define DOT11CURRENTCCAMODE                 0X04050103  // 4.5.1.3      1
```

```
#define DOT11EDTHRESHOLD                    0X04050104  // 4.5.1.4      4

#define DOT11REGDOMAINSSUPPORTINDEX         0X04070101  // 4.7.1.1      1
#define DOT11REGDOMAINSSUPPORTVALUE         0X04070102  // 4.7.1.2      1

#define DOT11PHYTYPE                        0X04010101  // 4.1.1.1      1
#define DOT11CURRENTREGDOMAIN               0X04010102  // 4.1.1.2      1
#define DOT11TEMPTYPE                       0X04010103  // 4.1.1.3      1

#define DOT11SUPPORTEDDATARATESTXINDEX      0X04090101  // 4.9.1.1      1
#define DOT11SUPPORTEDDATARATESTXVALUE      0X04090102  // 4.9.1.2      1

#define DOT11SUPPORTEDDATARATESRXINDEX      0X040A0101  // 4.10.1.1     1
#define DOT11SUPPORTEDDATARATESRXVALUE      0X040A0102  // 4.10.1.2     1

//
//  Local MIB Ids
//
#define CW10TALLIES                         0XFF000001  //              176

//
//  Status values define
//
#define DOT11_SUCCESS                       0x00
#define DOT11_FAILED                        0x01
```

B.3 Unimplemented OIDs

Some of the documented OIDs are not implemented in the driver. Until they are implemented, the documentation should be considered preliminary and subject to change. Unimplemented OIDs are:

- Query of `OID_CW10_GET_TALLIES` not implemented in driver
- Set of `OID_CW10_CLEAR_TALLIES` not implemented in driver
- `CW10_LOCAL_TALLY_STRUCT` not defined in the released include files

B.4 Obsolete OIDs

In previous releases of the PRISM Driver DDK, OIDs for both SET and GET were programmed as the same numeric value. The driver would differentiate between the functions based on which OID interface was called, Query vs. Set. Because the `MACIOCTL` does everything through the QUERY interface, we had to differentiate between the two actions with different OIDs. The old OID is retained, renamed to indicate it is a GET_SET operation and that it is now obsolete. The new GET OIDs will only work through the QUERY and the new SET OIDs will only work through either the QUERY or the SET interfaces.

For example, `OID_CW10_GET_SET_RID_OBSOLETE` will only work if you go through the SET interface for sets and the QUERY interface for GET. The new `OID_CW10_GETRID` will only work through the QUERY and `OID_CW10_SETRID` will work through the QUERY and SET interface.

Obsolete OIDs are recognized by Query and Set interfaces in order to allow software using earlier revisions of the driver to continue functioning. Supported obsolete OIDs are as listed below as defines and are grouped with the redefined OIDs that they previously supported (see also `CW10UIF.H`).

```
#define OID_CW10_GET_SET_RID_OBSOLETE        0xFF010181
#define OID_CW10_GETRID                      0xFF030001
#define OID_CW10_SETRID                      0xFF030002

#define OID_CW10_GET_SET_PDA_OBSOLETE        0xFF010182
#define OID_CW10_GET_PDA                     0xFF030003
#define OID_CW10_SET_PDA                     0xFF030004

#define OID_GET_SET_DOT11_MIB_OBSOLETE       0xFF020191
```

```
#define OID_GET_DOT11_MIB                        0xFF030007
#define OID_SET_DOT11_MIB                        0xFF030008

#define OID_CW10_READ_WRITE_AUX_OBSOLETE         0xFF010186
#define OID_CW10_READ_AUX                        0xFF030005
#define OID_CW10_WRITE_AUX                       0xFF030006

#define OID_CW10_GET_SET_TALLIES_OBSOLETE        0xFF02018D
#define OID_CW10_GET_TALLIES                     0xFF030018
#define OID_CW10_SET_TALLIES                     0xFF030019

#define OID_NETWORK_TYPE_OBSOLETE                0xFF010194
#define OID_GET_NETWORK_TYPE                     0xFF030009
#define OID_SET_NETWORK_TYPE                     0xFF03000A

#define OID_ETHERNET_TRANSLATION_MODE_OBSOLETE   0xFF02019B
#define OID_GET_ETHERNET_TRANSLATION_MODE        0xFF03000B
#define OID_SET_ETHERNET_TRANSLATION_MODE        0xFF03000C

#define OID_WEP_REQUIRED_MODE_OBSOLETE           0xFF02019C
#define OID_GET_WEP_REQUIRED_MODE                0xFF03000D
#define OID_SET_WEP_REQUIRED_MODE                0xFF03000E

#define OID_TX_RATE_OBSOLETE                     0xFF02019D
#define OID_GET_TX_RATE                          0xFF03000F
#define OID_SET_TX_RATE                          0xFF030010
```

Appendix C Example PDA

The following is an example of the PDA released for the EVB2 development NIC initialized for the North American regulatory domain:

```
5,8,0x8000,0,1,0                  // EVB2
6,7,0,2,1,1,1                     // Controller Supplier Range
4,0x101,0x2000,0x00CA,0x1913      // MAC Address
7,0x103,2,0x2010,0,0,0,0          // regulatory domain
2,0x104,0x07ff                    // Channel Set (default for NA)
2,0x105,3                         // default Channel (NA)
2,0x107,1                         // Temperature type
2,0,0                             // end of pda
```

The download utility `flash.exe` has an option to take a data file defining the PDA and download it into a new card (see the *PRISM Flash Utility User's Guide*). This utility takes the data file and converts it into a little-endian format binary image. The above example PDA would have the following format:.

PD Record	**Length**		**Code**		**Body**										
NicID	05	00	08	00	00	80	00	00	01	00	00	00			
ContSupRange	06	00	07	00	00	00	02	00	01	00	01	00	00	00	
MacAddr	04	00	01	01	00	20	CA	00	13	19					
RegDomain	07	00	03	01	02	00	10	20	00	00	00	00	00	00	00
ChanSet	02	00	04	01	FF	07									
DefChan	02	00	05	01	03	00									
TempType	02	00	07	01	tt	tt									
PDAEnd	02	00	00	00	crc	crc									

Appendix D Card Information Structure (CIS)

The CIS is a structure that must be present on PC Cards to convey certain information about the card to the software in direct control of the PCMCIA interface. The following table defines which software will interpret the information stored in the CIS.

NIC Operating Environment	Software components interpreting CIS
Client Station, Windows 95	Windows
Client Station, Windows NT	Windows
Client Station, Windows 3.11	Card & Socket Services (CSS) or Card Access Driver (CAD)
Client Station, DOS	Card & Socket Services (CSS) or Card Access Driver (CAD)
Server Station, Netware *nnn*	not supported

The CIS has a general structure of Type/Length/Value records. In CIS terminology, these records are called *tuples*. Detailed definition of the CIS tuples is provided in Reference [4].

For ease of use, the CIS tuples used by the PRISM MAC Host Driver are specified in this appendix. Where descriptive detail is missing, refer to Reference [4] (Volume 4, Metaformat Specification, Feb 1995).

The Compact CIS in the Flash ROM Memory Map

The CIS that is produced based on this specification is placed at a fixed location in the memory map of the Flash ROM on the PRISM MAC Host Driver NIC.

Start address	Maximum Length
003EFE00	240 bytes

The Expanded CIS in Attribute Memory

The CIS is accessible by the Host through the attribute address space. The PRISM MAC Firmware loads the CIS from its Flash ROM memory in the attribute memory after a reset. This load process also involves an expansion such that each byte of the CIS will show up in the low-order byte of a separate word in attribute memory.

D.1 PRISM MAC Host Driver Release 1 CIS Tuples

The CIS for the PRISM MAC Host Driver release 1 contains the following tuples in the order specified:

DEVICE
DEVICE_A
DEVICE_OC
DEVICE_OA
VERS_1
MANFID
FUNCID
FUNCE (6 times)
CONFIG
CFTABLE_ENTRY
END

D.1.1 NO_LINK TUPLE (NOT INCLUDED)

This tuple is removed because it creates problems in certain Notebook PCs, and is not strictly required.

D.1.2 DEVICE TUPLE

This device information tuple specifies common memory for 5V operation. Since the PRISM MAC has no common memory and the Boot ROM is not a functional part of the NIC, a NULL device is specified.

Byte	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	TPL_CODE: CISTPL_DEVICE = 01h							
1	TPL_LINK = 03h							
2	Device Type Code = 0 (NULL)				WPS = 0	Device Speed = 0 (NULL)		
3	00h (Device Size field not used)							
4	FFh (End of Device Info fields)							

D.1.3 DEVICE_A TUPLE

This device information tuple specifies the attribute memory supported by the PRISM MAC for 5V operation. The specified Device Speed is for write cycles only. For read cycles, the Host always assumes a 300ns cycle time.

Byte	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	TPL_CODE: CISTPL_DEVICE_A = 17h							
1	TPL_LINK = 03h							
2	Device Type Code = 6 (SRAM)				WPS = 0	Device Speed = 2 (200ns)		
3	Device Size = 08h (0–3FFh or 1k address range)							
4	FFh (End of Device Info fields)							

D.1.4 DEVICE_OC TUPLE

This device information tuple specifies common memory for 3.3V operation. Since the PRISM MAC has no common memory and the Boot ROM is not a functional part of the NIC, a NULL device is specified. Byte 2 is the Other Condition Info field.

Byte	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	TPL_CODE: CISTPL_DEVICE_OC = 1Ch							
1	TPL_LINK = 04h							
2	0	0	0	0	Vcc = 01 (3.3V)		0	0
3	Device Type Code = 0 (NULL)				WPS = 0	Device Speed = 0 (NULL)		
4	00h (Device Size field not used)							
5	FFh (End of Device Info fields)							

D.1.5 DEVICE_OA TUPLE

This device information tuple specifies the attribute memory supported by the PRISM MAC for 3.3V operation. The specified Device Speed is for write cycles only. For read cycles, the Host always assumes a 600ns cycle time. Byte 2 is the Other Condition Info field.

Byte	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	TPL_CODE: CISTPL_DEVICE_OA = 1Dh							
1	TPL_LINK = 04h							
2	0	0	0	0	Vcc = 01 (3.3V)		0	0
3	Device Type Code = 6 (SRAM)				WPS = 0	Device Speed = 2 (200ns)		
4	Device Size = 08h (0–3FFh or 1k address range)							
5	FFh (End of Device Info fields)							

D.1.6 VERS_1 TUPLE

This tuple specifies level 1 version compliance to the PCMCIA standard (Reference [4]) and manufacturer information.

Byte	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	TPL_CODE: CISTPL_VERS_1 = 15h							
1	TPL_LINK = 50h							
2	TPLLV1_MAJOR = 05h							
3	TPLLV1_MINOR = 00h							
4	TPLLV1_INFO = 00h 00h 00h reserved space (30-bytes filled with 00h)							
81	FFh (End of chain)							

The TPLLV1_INFO field is filled with ISO-646 Alphabet strings separated by 00h bytes.

The string “Version *xx.yy*” indicates the CIS version:

xx = major version; only updated for non-interchangeable changes
yy = minor version; updated for each new version

Notice that the version string is not included in the part of the CIS that is used by certain operating systems to calculate whether a card is a new card type.

D.1.7 MANFID TUPLE

This tuple denotes the card manufacturer code specified by the OEM manufacturer (e.g., 0156h), part number (e.g., 02h) and revision (e.g., 01h):

Byte	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	TPL_CODE: CISTPL_MANFID = 20h							
1	TPL_LINK = 04h							
2	TPLMID_MANF (low byte) = 56h							
3	TPLMID_MANF (high byte) = 01h							
4	TPLMID_CARD (low byte) = 02h (Card Identifier)							
5	TPLMID_CARD (high byte) = 00h (Card Revision)							

The encoding of the Card Identifier (byte 4) is as follows:

01 = Reserved
02 = Reserved
others = to be assigned

The encoding of the Card Revision (byte 5) is as follows:

00 = initial release of a new Card Identifier family
01 = first non-interchangeable change will increase revision.

Notice that it is important to keep this tuple constant in case there is an interchangeable change. This is important because this tuple is included in a checksum algorithm used by some operating systems to decide whether to treat a NIC as a new type of card or as an existing and known card type.

D.1.8 FUNCID TUPLE This tuple identifies the card as a Network Adapter:

Byte	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	TPL_CODE: CISTPL_FUNCID = 21h							
1	TPL_LINK = 02h							
2	TPLFID_FUNCTION = 06h (Network Adapter)							
3	TPLMID_SYSINIT = 00h							

D.1.9 FUNCE TUPLES In extension to the FUNCID tuple, a set of six FUNCE tuples specifies specific properties of the Network Interface Card.

The following FUNCE tuple specifies the card as a Wireless Network Adapter:

Byte	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	TPL_CODE: CISTPL_FUNCE = 22h							
1	TPL_LINK = 02h							
2	TPLFE_TYPE: LAN_TECH = 01h							
3	LAN_TECH_CODE = 07h (Wireless)							

The following two FUNCE tuples define the available bit rates of 1, 2, 5.5, and 11 Mbit/s.

Byte	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2b	bit 1	bit 0
0	TPL_CODE: CISTPL_FUNCE =22h							
1	TPL_LINK =05h							
2	TPLFE_TYPE: LAN_SPEED = ?							
3-6	LAN_SPEED_VALUE ?							

Byte	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2b	bit 1	bit 0
0	TPL_CODE: CISTPL_FUNCE =22h							
1	TPL_LINK =05h							
2	TPLFE_TYPE: LAN_SPEED = 02h							
3-6	LAN_SPEED_VALUE 2,000,000 (32-bit integer)							

The following FUNCE tuple specifies the card as a wireless LAN using a 2.4GHz spread-spectrum radio:

Byte	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	TPL_CODE: CISTPL_FUNCE = 22h							
1	TPL_LINK = 02h							
2	TPLFE_TYPE: LAN_MEDIA = 03h							
3	LAN_MEDIA_CODE = 07h (Spread-spectrum radio 2.4GHz)							

The following tuple defines the MAC address. The MAC address is product specific and is programmed during production. The MAC address in this Node ID tuple is the same as the default MAC address used by the NIC.

Byte	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	TPL_CODE: CISTPL_FUNCE = 22h							
1	TPL_LINK = 08h							
2	TPLFE_TYPE: LAN_NID = 04h							
3	LAN_NID_SZ = 06h (MAC address contains 6-bytes)							
4-9	LAN_NID_CODE = MAC address, low byte first.							

The following FUNCE tuple specifies the card connector type as closed connector:

Byte	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	TPL_CODE: CISTPL_FUNCE = 22h							
1	TPL_LINK = 02h							
2	TPLFE_TYPE: LAN_CONN = 05h							
3	LAN_CONN_CODE = 01h (Closed connector)							

D.1.10 CONFIG TUPLE

The configuration tuple specifies the base address of the COR register and the Configuration index to be written in the COR register. The COR register is the only function configuration register supported by the NIC.

Byte	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	TPL_CODE: CISTPL_CONFIG = 1Ah							
1	TPL_LINK = 07h							
2	0	0	TPCC_RMSZ = 0			TPCC_RASZ = 3		
3	0	0	Configuration Index = 01h					
4-7	COR register base address = 000003E0h (4-bytes, low byte first)							
8	TPCC_RMSK = 01h (COR register only)							

D.1.11 CFTABLE_ENTRY TUPLE

This tuple specifies the configuration table for 5V supply only operation configuration.

Byte	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	TPL_CODE: CISTPL_CFTABLE_ENTRY = 1Bh							
1	TPL_LINK = 0Eh							
2	IF = 1	Def = 1	Configuration Index = 01h					
3	TPCE_IF = 01h (Interface Description field)							
4	TPCE_FS (Feature Selection byte)							
	Misc 0	Mem Space 00		IRQ 1	IOSpace 1	Timing 0	Power 01 (Vcc only)	
5	TPCE_PD (Parameter Selection byte in Power Description)							
	0	Pdwn I 1	Peak I 1	Avg I 1	Static I 0	Max V 1	Min V 1	Nom V 0
6-7	Minimum Voltage = C5h, 4Bh (4.75V)							
8-9	Maximum Voltage = D5h, 19h (5.25V)							
10	Average Current = 36h (300mA)							
11	Peak Current = 36h (300mA)							
12	Power Down Current = 05h (10mA)							
13	TPCE_IO (I/O space description)							
	0	10 (16 bit I/O)		# of IO address lines = 6				
14	TPCE_IR (Interrupt Request description structure)							
	Share 0	Pulse 1	Level 1	Mask 1	VEND 1	BERR 1	IOCK 1	NMI 1
15	FFh (Interrupt mask 1)							
16	FFh (Interrupt mask 2)							

The PRISM MAC Host Driver will support both 5V and 3.3V operation. The tuple then becomes:

Byte	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	TPL_CODE: CISTPL_CFTABLE_ENTRY = 1Bh							
1	TPL_LINK = 0Eh							
2	IF = 1	Def = 1	Configuration Index = 01h					
3	TPCE_IF = 01h (Interface Description field)							
4	TPCE_FS (Feature Selection byte)							
	Misc 0	Mem Space 00		IRQ 1	IOSpace 1	Timing 0	Power 01 (Vcc only)	
5	TPCE_PD (Parameter Selection byte in Power Description)							
	0	Pdwn I 1	Peak I 1	Avg I 1	Static I 0	Max V 1	Min V 1	Nom V 0
6	Minimum Voltage = 35h (3.0V)							
7-8	Maximum Voltage = D5h, 19h (5.25V)							
9	Average Current = 36h (300mA)							
10	Peak Current = 36h (300mA)							
11	Power Down Current = 05h (10mA)							
12	TPCE_IO (I/O space description)							
	0	10 (16 bit I/O)		# of IO address lines = 6				
13	TPCE_IR (Interrupt Request description structure)							
	Share 0	Pulse 1	Level 1	Mask 1	VEND 1	BERR 1	IOCK 1	NMI 1
14	FFh (Interrupt mask 1)							
15	FFh (Interrupt mask 2)							

D.1.12 END TUPLE

This is the last tuple in the CIS:

Byte	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
0	TPL_CODE: CISTPL_END = FFh							

D.1.13 MAC ADDRESS IN FUNCE NODE-ID TUPLE

A unique universal MAC Address is assigned to each PRISM MAC NIC as it leaves the production facility. This MAC address must also appear in the FUNCE tuple defining the Node-Id.

Production Data Item	Pluggable Length
Plugdata.MACAddr (0101h)	6-bytes

Appendix E Frequently Asked Questions (FAQs)

1. **What is the amount of time taken per byte to transfer data over the PCMCIA interface?**

   Peak is 3 MCLKs (internal clock of the HFA3841) plus synchronization delay, which at 11MHz (making MCLK = 91 nanoseconds) is approximately 300 nanoseconds per word (since most transfers will be word transfers). However, there are other setup delays and buffer chaining delays which add 3 MCLK every 124-bytes. Also, the WEP engine will add an additional MCLK to each word transfer if it is operating during buffer transfers.

2. **Do you support PCMCIA DMA?**

   No. At the time we did the analysis (1995), the implementation of DMA on most socket controllers was slower than doing programmed IO transfer.

3. **What is the typical busy time for Data offset?**

   In the best case, there are 90 context 3 instructions that complete this function. With an 11MHz clock, this works out to 8.1 μsec from writing the offset register until the busy bit is cleared. This time can be extended if the processor has to handle higher level events (receives, etc.), but it's still going to be pretty fast.

4. **What is the typical BAP read/write cycle (i.e., how many words to read vs. setting the offset)?**

   The offset setting takes (best case) 8.1 μsec.

5. **For a Tx, when do you get the command complete (compared to the Tx complete)?**

   The command complete takes about 140 μsec. The start of the transmit takes about 70 μsec from command complete (regardless of the size of the packet, since the firmware is only moving around pointers to buffers). Transmit Complete depends on a number of factors, such as how large the packet is and the number of retries.

6. **At what point do you receive the next Rx packet if there are two waiting? Just after the ACK? How many μsec?**

   Receive indications are queued, so when you acknowledge the Rx event and then read the EvStat register, the next Rx event will be indicated. If you wait for an event, the latency will be determined by the host. We recommend reading the EvStat register after each acknowledge.

7. **What happens if I ACK the Rx, but both BAPs are being used for Tx?**

   You won't ACK the received packet, the firmware will. You will get a Rx event indication (or rather the HFA3841 hardware will assert the event) no sooner than approximately 115 μsec after the receipt of a valid CRC of the frame. If you are using both BAPs for transmits then you will just have to wait for one to be free to retrieve the received packet. Note that the packet is still available to you until you acknowledge the Rx event. If you acknowledge the Rx event before reading out the data through a BAP, the packet is discarded.

8. **Can I just collect the Receive FIDs and do the transfer when I am able?**

   You can only retrieve the first receive FID from the RxFID register. The next receive FID won't be available in the RxFID until the previous receive event is acknowledged.

9. **How are Broadcasts in PowerSave handled (when all STAs switch from ps to nonps)?**

   If cnfMulticastPMBuffering (rid FC17) is set, then multicast frames will be queued on a special multicast PowerSave queue to be sent on DTIMs. Broadcast/multicast frames are queued until the DTIM.

10. **You have a PowerSave switch in the IF. Is this not intended for use in the AP?**

    The API (rids FC09, FC0A, FC0B, FC0C, FC0D) is for the station.

11. **In PowerSave, do you think we need to throttle broadcasts and unicasts based on what state the AP is in (in other words, Broadcasts only go out after a DTIM)?**

    The host can throttle the number of outstanding transmits by tracking the frames sent to the PRISM MAC with the transmit complete frames indicated by the PRISM MAC. This way the host can do its own PRISM MAC queue depth manipulation. This should be equally effective for unicasts and multicasts.

12. **How do we know when to turn off PS management of broadcasts?**

    The PowerSaveUserCount notification message will be automatically generated by the firmware when the count transitions between 0 and 1 (and back to 0).

13. **In the previous API document there is a sentence that reads something like “the More Data subfield is copied unaltered to the from the host to the CHOICE MAC for transmission.” Why? Is this correct?**

    If you wanted to do the AP functions in the Host then the More Data subfield (and others) should not be modified by the firmware when transmitting the packet.

14. **Is there a way to override the type and subtype fields to use an AP to spoof a station?**

    The host driver can send any type of frame through the packet transmit API. The host needs to specify that it is using the 802.11 header instead of the 802.3 header.

15. **Which length is byte swapped, the 802.11 or 802.3? Is the CHOICE little-endian or big-endian?**

    There are two length fields in the PRISM MAC packet header. There is a length in the 802.11 portion of the header and one in the 802.3 portion. The 802.11 length (and the CHOICE) is little-endian and the 802.3 length is big-endian.

16. **What are the true timing numbers? The min/max/aver time for 64-byte/1518-byte/ 2334-byte packet from the time the transmit command is entered to the time it is to be transmitted over the air?**

    We have performed many timing tests but have not completely characterized timing in all situations. We do have some preliminary numbers for the transmit command timeline:

Time till command accepted (or complete):	~144 µsec
Time till the packet starts transmitting:	~70 µsec
Alloc Event for reclaimed FID:	~98 µsec

These are best case numbers. When you start loading down the systems you will see the allocation event occur before the start of transmit because of internal queuing of packets.

E.1 Station Auto Rate Fallback

The Station Auto Rate Fallback model controls the current transmit rate of the packet. Starting from the highest rate of the association, the fallback algorithm is triggered by the transmit failure of a frame. Transmit failures occur when all retries have been sent without acknowledgment. The trigger of the RateFallbackDuration timer or a number of successful transmissions controls upgrade back to the highest rate.

The association’s supported rates is an AND operation between the rates reported by the AP (in association response) and the rates configured by the Station Host in TxRateControl RID.

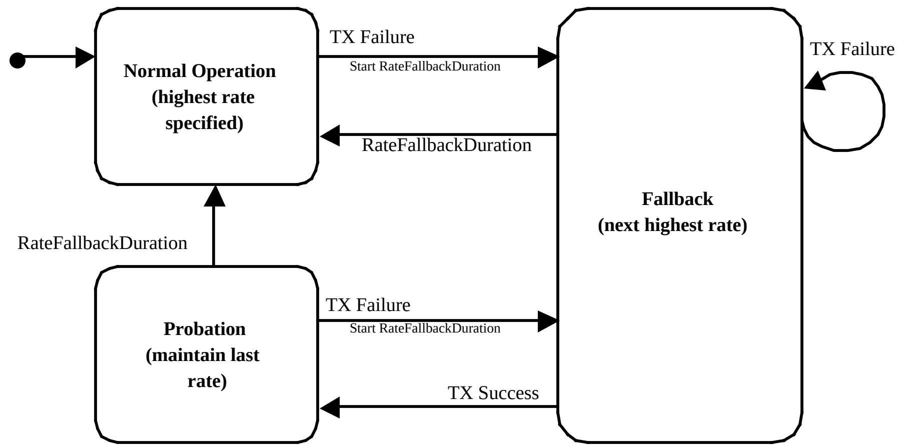


Figure F.1: Station Auto Rate Fallback

E.1.1 STATES

Normal Operation – This state indicates normal operation at the highest rate of the association.

Fallback – This state represents the operation when the station is dropping back through the specified rates. This state is entered when frame transmits are reported as failed due to retry limit exceeded.

Probation – This state is “on the way back” to the normal full speed operation.

E.1.2 EVENTS

TX Failure – This event indicates that a frame was not transmitted successfully.

RateFallbackDuration – This event indicates that the timer used to measure a continuous period in fallback state has expired. The value for RateFallbackDuration is fixed at 10 * 1000Kus (10 seconds).

Appendix F Firmware Test Commands

Special test firmware gives hardware engineers the ability to test the modem and enables manufacturing testing. Some of the tests can be performed with the standard PRISM MAC firmware, but the remainder must be written since there is no other way to test that component.

F.1 Description

Firmware for engineering test is accessible through special test mode commands. Those test functions that are currently implemented are part of the standard secondary firmware image. The test firmware may reside in a stand-alone firmware image in the future.

These PRISM test commands are used together with the standard operational commands, such as Allocate, ReadMIF and WriteMIF (see 4.2 Interface Commands).

Using the downloaded firmware, all the special firmware components are started with a dedicated PRISM test command. The PRISM test mode command has the following format:

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	Busy	0	0	Test mode					0	0	Command code = 38 (hex)					

F.2 Test Modes (TM)

The following are the test modes for the PRISM test command:

- 0 – Reset PHY using PHY services
- 1 – Init PHY using PHY services
- 2 – PHY sleep using PHY services
- 3 – PHY wake-up using PHY services
- 8 – Change channel / frequency
- 9 – Suppress Post back-off delay
- A – Suppress Tx Exception
- B – Start monitor mode
- C – LED test
- E – Start continuous transmission; data (16-bits) specified in parameter 0
- F – Stop Testing
- 10 – Continuous Receive
- 11 – Set Signal
- 13 – Cal Enable
- 15 – Configuration Bits control

The test modes 9, A, B, C, E, and 10 simply start a test mode. Stopping any of these modes must be done by test mode F.

Following is a detailed description of the different modes:

F.2.1 TM0: RESET PHY

The firmware uses PHY services to reset the PHY and the Cmd event is set.

F.2.2 TM1: INIT PHY

The PHY is powered on, reset and the PRISM™ and synthesizers are fully configured by PHY services.

F.2.3 TM2: PHY SLEEP

The PHY is placed in sleep mode PHY services. No other actions are performed by the firmware.

F.2.4 TM3: PHY WAKE-UP

The PHY awakes by PHY services and the Cmd event is set.

F.2.5 TM8: CHANGE CHANNEL

PHY services are used to change to a numbered channel. This firmware writes the required channel information from the built-in channel list into the Prism radio. This command returns after a delay of 250 µs. The required channel (1 – 14) must be written in the Param0 register, prior the invocation of this test.

NOTE: No check is made on channel number, so a channel number of 0 or 15 or greater can give an unpredictable result.

NOTE: If the port mode (RID FC00) is BSS (perhaps others) then the MAC STA-Firmware may change the channel back to an available channel, which it can associate to an AP.

F.2.6 TM9: POST BACK-OFF DELAY SUPPRESSION

When set to this mode the "post back-off delays", which occur after a transmitted frame, are suppressed to obtain a better throughput.

F.2.7 TMA: TRANSMIT EXCEPTION SUPPRESSION

When set to this mode any transmit frame will return a correct result, regardless of whether the frame was acknowledged correctly.

F.2.8 TMB: MONITOR MODE

The firmware starts receiving in monitor mode. In this mode a station receives and passes to the host all frames wherein the PLCP header and beginning of the frame are successfully received, regardless of FCS errors or other protocol defects. All received frames are passed to the host, and no check is made on destination address, source address, FCS result, or WEP decryption result. Encrypted frames are decrypted.

Note that this mode differs in significant aspects from Promiscuous Mode (see 4.4.2.2.17 PromiscuousMode).

F.2.9 TMC: LED TEST

This command tests the Power and Activity LEDs. Param0 indicates which LED to test, where a value of 1 for bit 0 indicates the Power LED, and a value of 1 for bit 1 indicates the Activity LED.

Param1 specifies the rate of the test., that is, the frequency at which the LED is turned on and off. A value of 50 tests at a rate of 10 KHz (50 µsec on and 50 µsec off). A value of -1 tests at a rate of approximately 8 Hz

F.2.10 TME: CONTINUOUS TRANSMIT

The firmware starts sending a continuous stream to facilitate manufacturing adjustments of the modem. The transmission of the continuous preamble must be stopped with the Stop command.

Command: 0e38

Parameter: Repetitive Transmission Data (MPDU) Pattern

This command provides a mechanism for placing the radio into a continuous transmission state. When invoking the command in Testware the 16-bit parameter supplied is used as a repetitive data pattern during transmission (i.e., a parameter of a010 would result in an MPDU that looks like 1010000000010000101000000000010000...). This command must be terminated by the Stop Test command (0f38 without parameters). If a new transmit sequence is desired, you must terminate the current Continuous TX (0f38) and then invoke again with the new transmit sequence parameter.

F.2.11 TMF: STOP TEST

Command: 0f38

Parameter: None

This command stops the current test in an elegant manner. This command must be used to terminate all of the commands that don't automatically stop.

This command reads the ID of the PRISM II/PRISM MAC hardware and writes it in the low byte of the Resp2 register.

The firmware stops the following tests:

Test Mode	
9	Suppress Post back-off delay
A	Suppress Transmit Exception
B	Monitor mode
E	Continuous preamble transmission
10	Continuous Receive

F.2.12 TM10: CONTINUOUS RECEIVE

Command: 1038

Parameter: None

This command invokes a continuous receive state. If a transmitted packet is detected and MDRDY goes high, then a Stop Test command must be executed followed by a re-invocation of Continuous RX.

F.2.13 TM11: SET SIGNAL STATE

Command: 1138

Parameter: Signal Mask

This command provides a method for direct control of several radio signal states. Specifically, PA_PE, TXPE_IF, RXPE_IF, RXPE_BB, and RESET_BB are controllable. (More will be added later.) The following tables indicate the bit assignments:

3860B Baseband Processor Bit Assignments	
BIT	**SIGNAL NAME**
15	PA_PE
14	TXPE_IF
13	RXPE_IF
12	RXPE_BB
11	RESET_BB

3861 & 3863 Baseband Processors Bit Assignments	
BIT	**SIGNAL NAME**
15	TXPE
14	PAPE
13	RXPE
12	RESET_BB
11	LED
10	PE1
09	PE2
08	RADIO_PE
07	CAL_ENABLE
06	TR_SW

A 1 in a given field will set the specified signal. A 0 clears it. This particular command may be invoked repeatedly without terminating it with a Stop Test command, but when transitioning to any of the other commands a Stop Test command must be issued.

Warning! Be very careful when using this command. There is no error prevention capability. Therefore, an illegal signal state may be entered which may have the potential of damaging the radio.

F.2.14 TM13: CAL ENABLE

Command: 1338

Parameter: None

This command issues a CalEnable to the Baseband processor.

F.2.15 TM15: CONFIGURATION BITS CONTROL

Command: 1538

Parameters: Param0 contains the set (1) or clear (0) request. Param1 contains the bit(s) to be set or cleared.

The configuration bits control the operation of the firmware. The following is the bit assignments:

Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name													ALC			

The Automatic Level Control bit (bit 3) controls whether the firmware performs ALC on transmit frames. All other bits are reserved and should not be set or cleared.

To turn off ALC operation the host must perform the following sequence:

1. write 0 to Param0
2. write 0x0008 to Param1
3. write 0x1538 to CMD

To turn on ALC operation the host must perform the following sequence:

1. write 1 to Param0
2. write 0x0008 to Param1
3. write 0x1538 to CMD