1394 PRINTER WORKING GROUP

CONFIGURATION ROM
for
IMAGING DEVICE PROFILE

*** DRAFT PROPOSAL ***

Revision 0.2b - January 21, 1999

Author – Gregory A. LeClair
EMAIL: greg@erc.epson.com
TEL: 1.408.474.0500 x3610
FAX: 1.408.474.0511

1 Scope and Purpose

1.1 Scope

This document specifies the Configuration and Status Registers (CSR) and the Configuration ROM of an SBP-2 target node that implements the minimal requirements to support the 1394 PWG Imaging Device Profile. This profile includes elements from released standards and work in progress by other groups referenced in Section 3.

This document does not address:

- Isochronous communication.
- Use with 1394.1 bridges.
- Security.
- Power Management Issues.

1.2 Purpose

The 1394 PWG has focused on defining a general purpose solution which uses the ANSI SBP-2 protocol for peripheral devices, from simple single function devices to more complex compound devices. A simple device may support multiple services, with one host, via a single login. Compound devices which support different functionality are accessed via separate unit directories.

Such partitioning allows each functional device object to support multiple services, to one or multiple hosts (based on device resources), via a single login.

The purpose of this document is to define the CSR and Configuration ROM requirements for printers, scanners, copiers, digital still cameras and other imaging devices which support the 1394 PWG Imaging Profile.

Requirements are specified in conformance to applicable standards. In all areas that are mandatory, the applicable standards will apply. Where applicable standards allow more than one choice of implementation, this document defines either a choice or preference for the 1394 PWG Imaging Profile.

The term “image device” is used throughout the remainder of this document to refer to image devices in general including any of the devices listed above.

2 References

This document makes reference to and contains excerpts from several standards or draft documents of proposed standards.

More recent revisions may or may not support the information contained in this document:

1. ISO/IEC 13213:1994 Control and Status Register Architecture for Microcomputer Buses.
2. IEEE Std 1394-1995, Standard for High Performance Serial Bus.
3. ANSI T10/1155x (SBP-2) Serial Bus Protocol 2
4. IEEE-p1394a Draft Standard for a High Performance Serial Bus (Supplement).
5. IEEE-1212r Draft – Revision to ISO/IEC 13213:1994

3 Definitions and Notation

3.1 Definitions

3.1.1 Conformance

See SBP-2 Section 3.1.1.

3.1.2 Glossary

See SBP-2 Section 3.1.2.

3.1.3 Abbreviations

See SBP-2 Section 3.1.3

3.2 Notation

3.2.1 Numeric Values

See SBP-2 Section 3.2.1

3.2.2 Bit, Byte and Quadlet ordering

See SBP-2 Section 3.2.2

4

5

6

7 CSR Definitions

7.1 Requirements

Compliant devices shall implement Configuration and Status registers as required by ISO/IEC 13213:1994, IEEE-1212r, IEEE Std 1394-1995, and ANSI SBP-2. This profile does not define any additional registers.

7.1.1 TBD

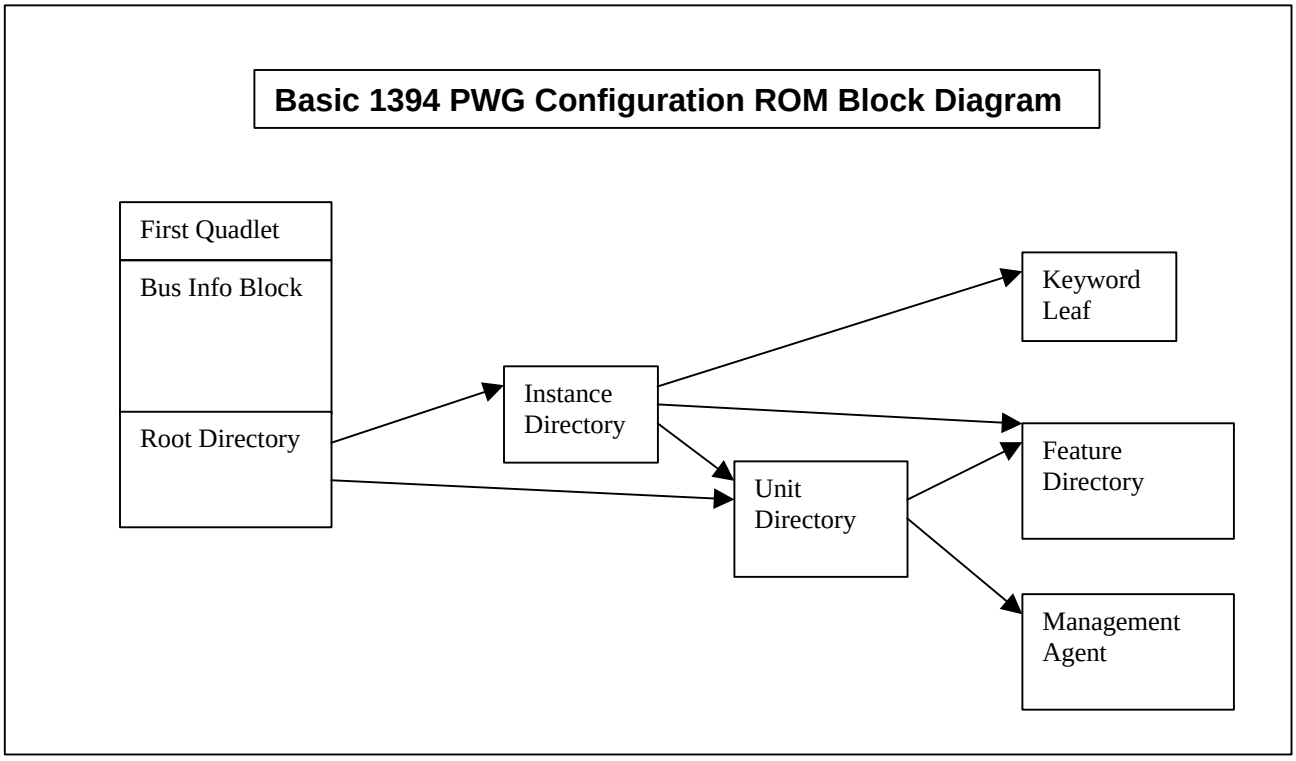
TBD

8 Configuration ROM

8.1 Requirements

Compliant devices shall implement general format configuration ROM in accordance with ISO/IEC 13213:1994, IEEE-1212r, IEEE Std 1394-1995, ANSI SBP-2, and this profile. The general format configuration ROM directory structure is an extensible self-describing hierarchy of information blocks.

The block diagram below illustrates a minimum set of Configuration ROM objects required by this profile.



The locations of the initial blocks, *Bus_Info_Block* and *Root_Directory*, are fixed. The locations of the other entries are specified in the *Root_Directory* or associated directories.

Note:
Reserved fields shall be set to zero.
Length values in the Configuration ROM specify the number of Quadlets.
There are two types of offsets specified by ISO 13213/IEEEE 1212.
1) Initial register space offset which is an offset in quadlets from the initial register space base address of 0xFFFF F000 0000. Value contained in the register multiplied by 4 plus base address.
2) Indirect space offset, which is an offset in quadlets from the current register address. Value contained in the register multiplied by 4 plus address of register.
Number 1 above has a key_type of 0x1. Number 2 above has a key_type of 0x2 or 0x3, see ISO 13213/IEEEE 1212 section 8.2.4 table 21 for all key_type definitions.

8.1.1 First Quadlet

Compliant devices will implement the first quadlet of configuration ROM as defined in the SBP-2 Draft. The first quadlet of configuration ROM is at the base address of FFFF F000 0400_{16}. A read of this location will indicate when the node completes initialization.

Compliant devices shall return a value of 0000 0000_{16} for any read request to FFFF F000 0400_{16} until the device is capable of supporting read transactions at other locations. The result of read transactions at other locations while this value is zero are unspecified.

After initialization, devices will return an implementation specific non-zero value.

8.1.2 Bus Information Block

Compliant devices shall implement the bus information block as defined in the SBP-2 Draft and the generate field bits as defined in IEEE p1394a.

Compliant devices shall return unique Chip_ID_High and Chip_ID_Low values. In conjunction with the Node_Vendor_ID, this provides an EUI-64 (Extended Unique Identifier, 64 bits).

If a bus node supports multiple units, then the EUI-64 must not be referential to any one unit directory to allow for unique identification of a unit in a multifunction device. The EUI-64 in the bus information block must be invariant when read with quadlet read requests.

8.1.3 Root Directory

Compliant devices shall implement the root directory immediately following the Bus_Info_Block as defined in the SBP-2 Draft.

Compliant devices shall implement the Module_Vendor_ID entry, the Textual_Descriptor_Offset entry, the Node_Capabilities entry, one or more Instance_Directory_offset entries and one or more Unit_Directory_offset entries.

8.1.4 Instance Directory

Compliant devices should / shall implement at least one instance directory as defined in the IEEE-p1212r Draft containing a Keyword leaf, one or more Feature Directory entries, zero or more Unit Directory entries, and zero or more Instance Directory entries.

8.1.5 Keyword Leaf

Compliant devices should / shall implement a Keyword leaf as defined in the IEEE-p1212r Draft.

8.1.6 Feature Directory

Compliant devices should / shall implement at least one instance directory as defined in the IEEE-p1212r Draft containing one or more valid Function_Class and Unit_Directory_offset entries.

NOTE: Is the Instance Directory, Keyword Leaf, and Feature Directory considered a mandatory requirement (shall) or strongly recommended (should)?

8.1.7 Unit Directory

Compliant devices shall implement at least one unit directory in the format specified by this profile. The unit directory shall contain Unit_Spec_ID and Unit_SW_Version entries as specified in ISO/IEC 13213:1994, a Management_Agent entry as specified by SBP-2, Cmd_Set_Spec_ID, Command_Set, Command_Set_Revision and Firmware_Revision entries as defined by this profile.

Compliant devices must support LUN 0 and at least one LU_Characteristics entry, one Logical_Unit_Number entry and one LUN Textual_Descriptor entry. The LUN Textual_Descriptor follows the format defined in IEEE-1284-1994 Section 6.6

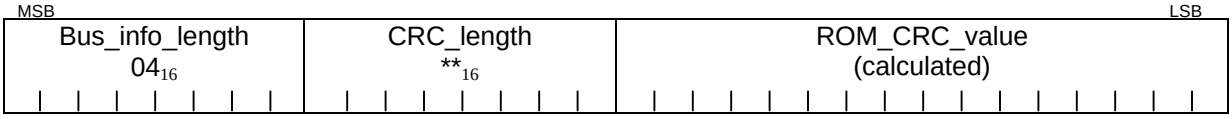
Logical Unit Directory structures should be implemented only if a node needs to define more than one Cmd_Set_Spec_ID, Command_Set, LU_Characteristics, Command_Set_Revision, or Firmware_Revision entry within a unit directory.

8.2 Sample

This section provides an example of the config ROM for a simple printing device.

8.2.1 First Quadlet

Offset: 0400_{16}



The first quadlet of configuration ROM is at the base address of FFFF F000 0400_{16}. A read of this location will indicate when the node completes initialization.

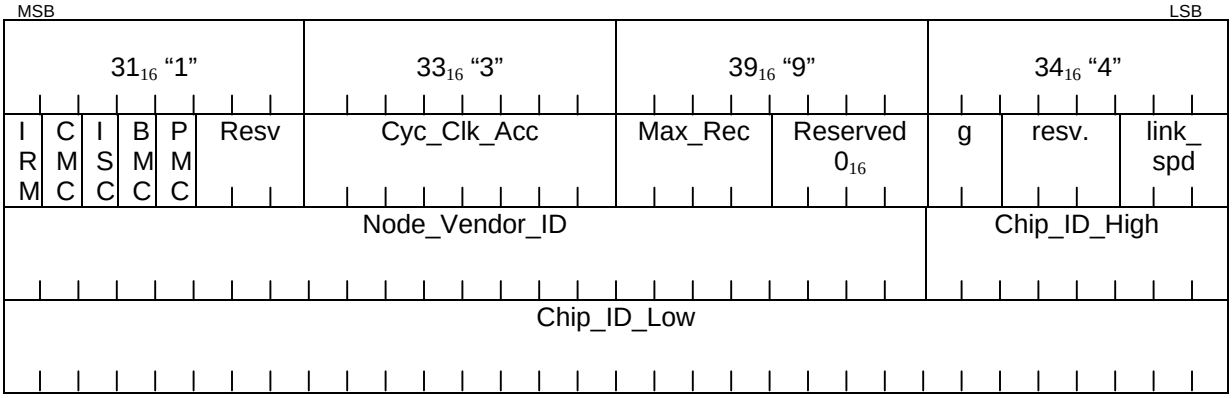
The bus_info_length value is 04_{16} for the Bus_Info_Block defined in this profile.

The CRC_length value is set to the number of quadlets to be used in calculating the ROM_CRC_value. The minimum value used for the CRC_length shall provide coverage for the Bus_Info_Block. The maximum CRC_length value provides coverage up to 255 quadlets.

The ROM_CRC_value is calculated according to the formula in ISO/IEC 13213:1994 Section 8.1.5.

8.2.2 Bus Information Block

Offset: 0404_{16}



Compliant devices shall implement the bus information block located at a base address offset of FFFF F000 0404_{16} in the format defined by this profile.

The first quadlet of the bus information block at offset 404h is the configuration ROM signature field used to identify an IEEE 1394 configuration ROM. This quadlet must contain the ASCII string "1394".

The second quadlet of the bus information block at offset 408h contains capability bits. The IRM, CMC and ISC bits and the Cyc_clk_acc field are required for nodes that support isochronous operation. The BMC bit indicates nodes that are bus manager capable. The PMC bit indicates nodes that are power manager capable.

The max_rec field defines the maximum data payload size supported by the node. The maximum_payload_size == 2^{max_rec+1} in bytes.

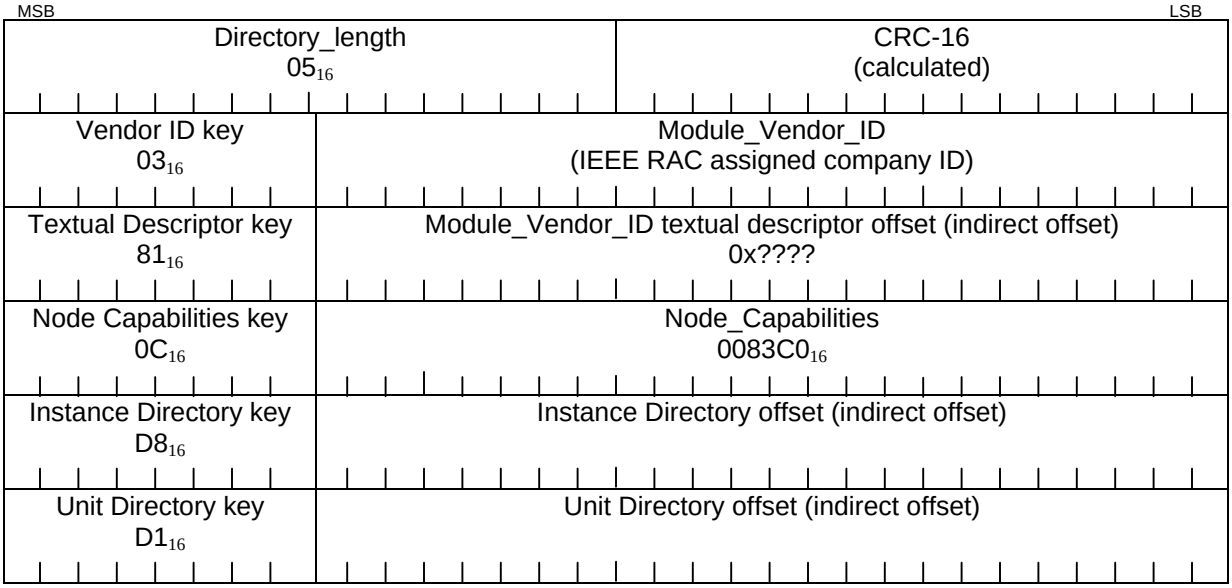
The g field bits affect the CRC calculation when the CRC_Length in the first quadlet of configuration ROM covers the Bus_Info_Block. These bits shall be changed when the device updates information in the configuration ROM that alters the device configuration. After the bits are changed, the device must recalculate the CRC stored in the first quadlet of configuration ROM. The changes to the generate bits and the ROM_CRC_Value shall be effected during a bus reset.

The third and fourth quadlets of the bus information block contain the Node_Vendor_ID, Chip_ID_High and Chip_ID_Low values. Chip_ID_High and Chip_ID_Low values should be unique between different units supplied by the same vendor. Together, these values provide a globally unique device ID. Because physical device addresses can change following a bus reset, this unique 64-bit node ID is the reliable method of node identification. This identifier is called the EUI-64 (Extended Unique Identifier, 64 bits).

If a bus node supports multiple units, then the EUI-64 must not be referential to any one unit directory to allow for unique identification of a unit in a multifunction device. The EUI-64 in the bus information block must be invariant when read with quadlet read requests.

8.2.3 Root Directory

Offset: 0414_{16}



Compliant devices shall implement the root directory located at a fixed address following the bus information block. As shown the root directory is located at a base address offset of FFFF F000 0414_{16}.

The first quadlet of the root directory contains directory_length and CRC-16 values. Each of these values is 16 bits in length.

The second quadlet contains the Module_Vendor_ID. The concatenated values of key type and key value for the Module_Vendor_ID is $03_{16.}$ The Module_Vendor_ID value should contain the 24 bit OUI of the manufacturer.

The third quadlet contains a pointer to a textual descriptor leaf. The concatenated values of key type and key value is $81_{16.}$ The textual descriptor should contain the name of the entity identified by the Module_Vendor_ID.

The fourth quadlet contains the Node_Capabilities entry. The concatenated values of key type and key value for the Node_Capabilities entry is $03_{16.}$ This contains subfields specified by ISO/IEC 13213:1994. Compliant devices will implement the SPLIT_TIMEOUT register, 64 bit fixed addressing scheme, the STATE_CLEAR.*lost* bit, and the STATE_CLEAR.*dreq* bit. Support for a capability is indicated by setting the appropriate bit to one. The value shown is an example and may be different between implementations. *(See section 8.4.11 of ISO/IEC 13213:1994 for more details on this entry.)*

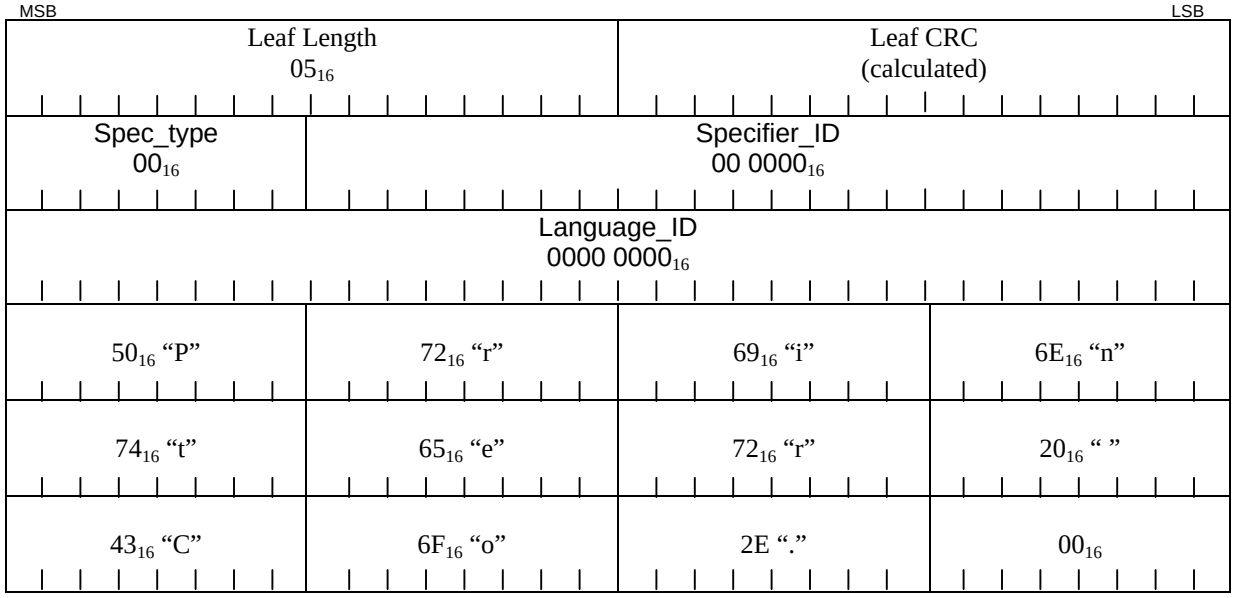
The fifth quadlet contains the Instance_Directory_offset entry. The concatenated values of key type and key value for the Instance_Directory_offset entry is $D8_{16.}$ The Instance_Directory_offset value is an offset to the instance directory for this node.

The sixth quadlet contains the Unit_Directory_offset entry. The concatenated values of key type and key value for the Unit_Directory_offset entry is $D1_{16.}$ The Unit_Directory_offset value is an offset to the Unit directory that implements the software interface for this node. *(As per current thinking in IEEE-1212r working group – this profile defines an implementation which should operate*

with legacy and future bus enumerations strategies. See section 8.5.3 of ROM-R00.pdf which is the current draft of the configuration ROM section provided by the 1212r group.)

8.2.4 Module_Vendor_ID_Textual_Descriptor

Offset: $042C_{16}$



Compliant devices shall implement a textual descriptor leaf that is referenced from the Root Directory. This profile minimally requires ASCII encoded textual descriptors. The textual descriptor shall contain the name of the company referenced by the Module_Vendor_ID entry in the Root Directory. The string 'Printer Co.' is used here as an example.

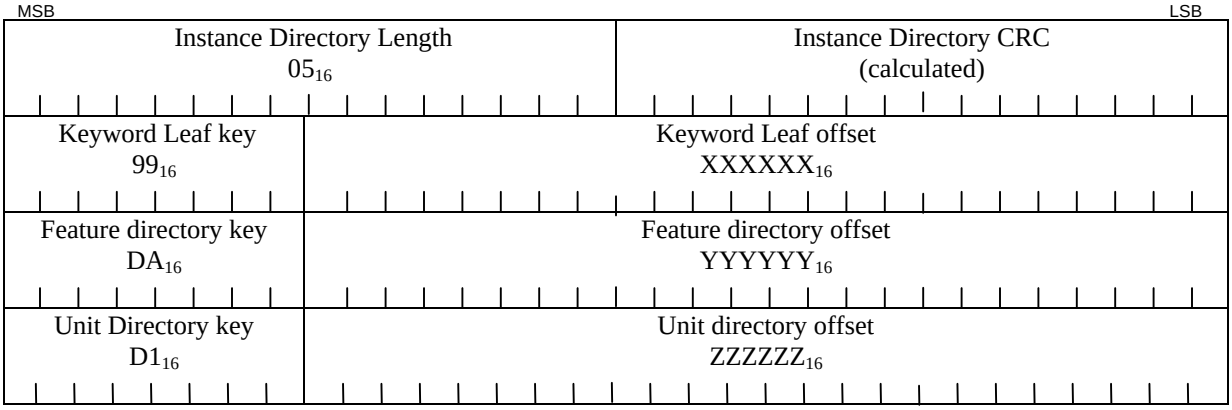
The first quadlet of the textual descriptor leaf contains the leaf_length and CRC-16 values. Each of these values is 16 bits in length.

The second and third quadlets contains the Spec_type, Specifier_ID, and Language_ID entries. The values for each of these fields is $00_{16,}$ 00 $0000_{16,}$ and $0000\ 0000_{16}$ respectively. This indicates an ASCII encoded string.

The remaining quadlets contains the ASCII encoded string as a sequence of non-zero characters. If the length, in bytes, of the text string is not a multiple of four, the last quadlet shall be padded with bytes whose value is 00_{16}.

8.2.5 Instance Directory

Offset: 0444_{16}



Compliant devices should / shall implement at least one Instance Directory located at an offset that is pointed to from the root directory. As shown the Instance Directory is located at a base address offset of FFFF F000 0444_{16}. An Instance Directory contains a Keyword leaf, one or more Feature Directory entries, zero or more Unit Directory entries, and zero or more Instance Directory entries.

The example shown above, for a simple device, provides for the Keyword leaf, one Feature Directory entry, and one Unit Directory entry.

The first quadlet of the instance directory contains the length and the CRC for the directory.

The second quadlet of the instance directory contains the Keyword Leaf offset. A keyword leaf is a collection of one or more ASCII keywords that provide information about the node which can be accessed via simple 1394 read transactions.

The third quadlet contains the Feature Directory offset. Feature directories provide an additional level of detail about this instance. Any unit directories included in the instance directory must support the feature set described in the Feature directory. This profile defines a PWG feature directory.

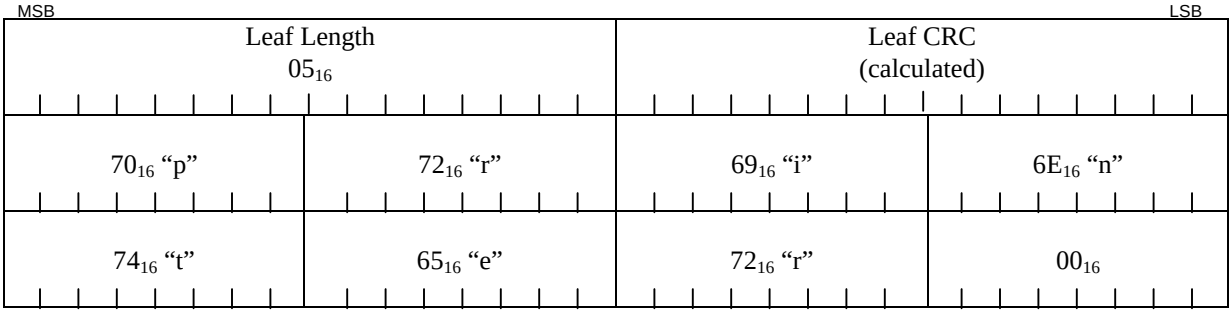
The fourth quadlet contains the Unit Directory offset. This is the software interface to the device. A Unit directory may also be pointed to from the root directory.

Note: The elements shown above are a minimal configuration for a compliant node. Vendors may implement various configurations of Instance directories, Feature directories, and Unit directories based on their design.

Issues:
1. Include keywords here or separate section?
2. How much information here about hierarchy, etc. Or just point to the 1212r draft docs.

8.2.6 Keyword Leaf

Offset: 0444_{16}



Compliant devices shall implement at least one Keyword Leaf located off of the Instance Directory. As shown the Keyword Leaf is located at a base address offset of FFFF F000 0444_{16}. A keyword leaf is a collection of one or more ASCII keywords that pertain to the parent directory which referenced the leaf.

A keyword leaf may be a child of the root, in which case it is likely to be a master index of all keywords present in all the keyword leaves in the device's configuration ROM. Or a keyword leaf may be a child of an instance directory, in which case its keywords describe the functions supported by that particular component.

Individual keywords within a keyword leaf shall be zero-terminated ASCII strings. The character set for keywords is an ASCII subset consisting of the characters 'a' through 'z' (uppercase is not allowed), '0' through '9' and the hyphen '-'; neither spaces nor any other characters, printing or nonprinting, shall appear in keywords.

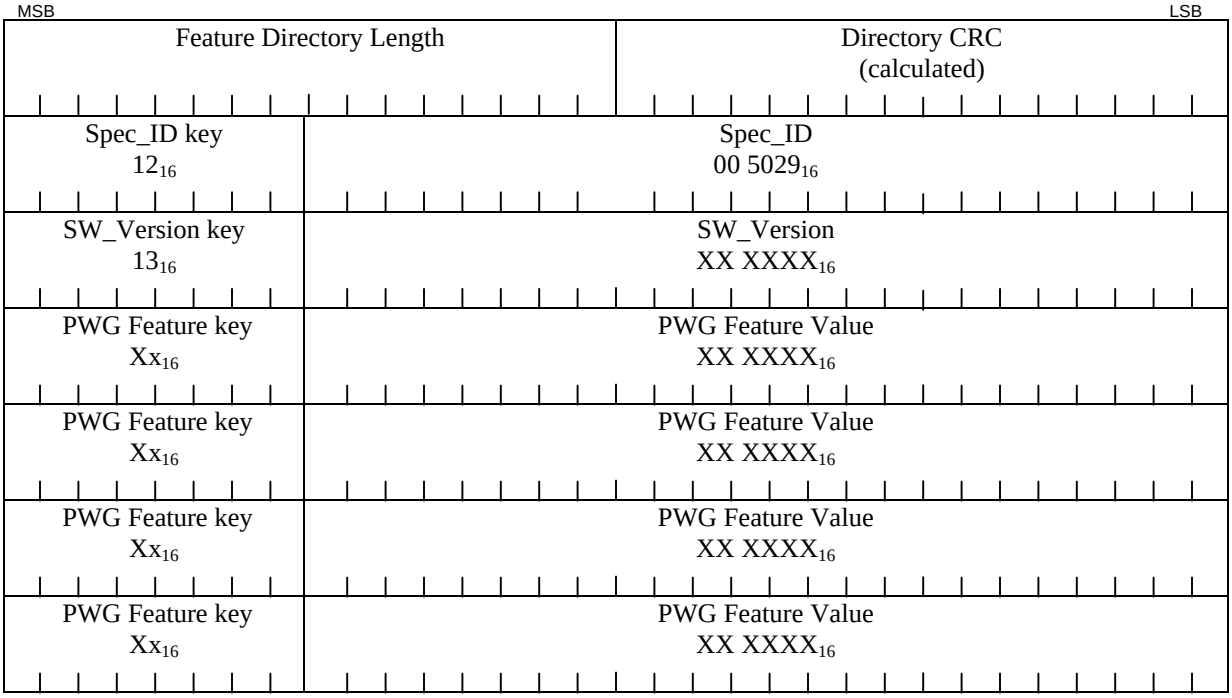
The example shown above, for a simple device, implements the keyword “printer”.

The first quadlet of the Keyword Leaf contains the length and the CRC for the leaf.

The remaining quadlets contain individual keywords encoded using the defined ASCII subset. Multiple keywords can be listed, they are separated by a single byte whose value is 00_{16}. If the length, in bytes, of the keywords and separators are not a multiple of four, the last quadlet shall be padded with bytes whose value is 00_{16}.

8.2.7 Feature Directory

Offset: 0444_{16}



Compliant devices shall implement at least one Feature Directory located at an offset that is pointed to from an Instance Directory and the associated Unit Directory. As shown the Feature Directory is located at a base address offset of FFFF F000 $xxxx_{16}$. A Feature Directory contains a collection of feature entries. The feature entries are to be interpreted based on the Spec_ID and SW_Version fields in the directory.

The example shown above, for a simple device, provides for a list of features to be defined by the PWG. Implementations may provide for one or more feature directories. For example, devices may implement the PWG and a vendor specific feature directory.

The first quadlet of the feature directory contains the length and the CRC for the directory.

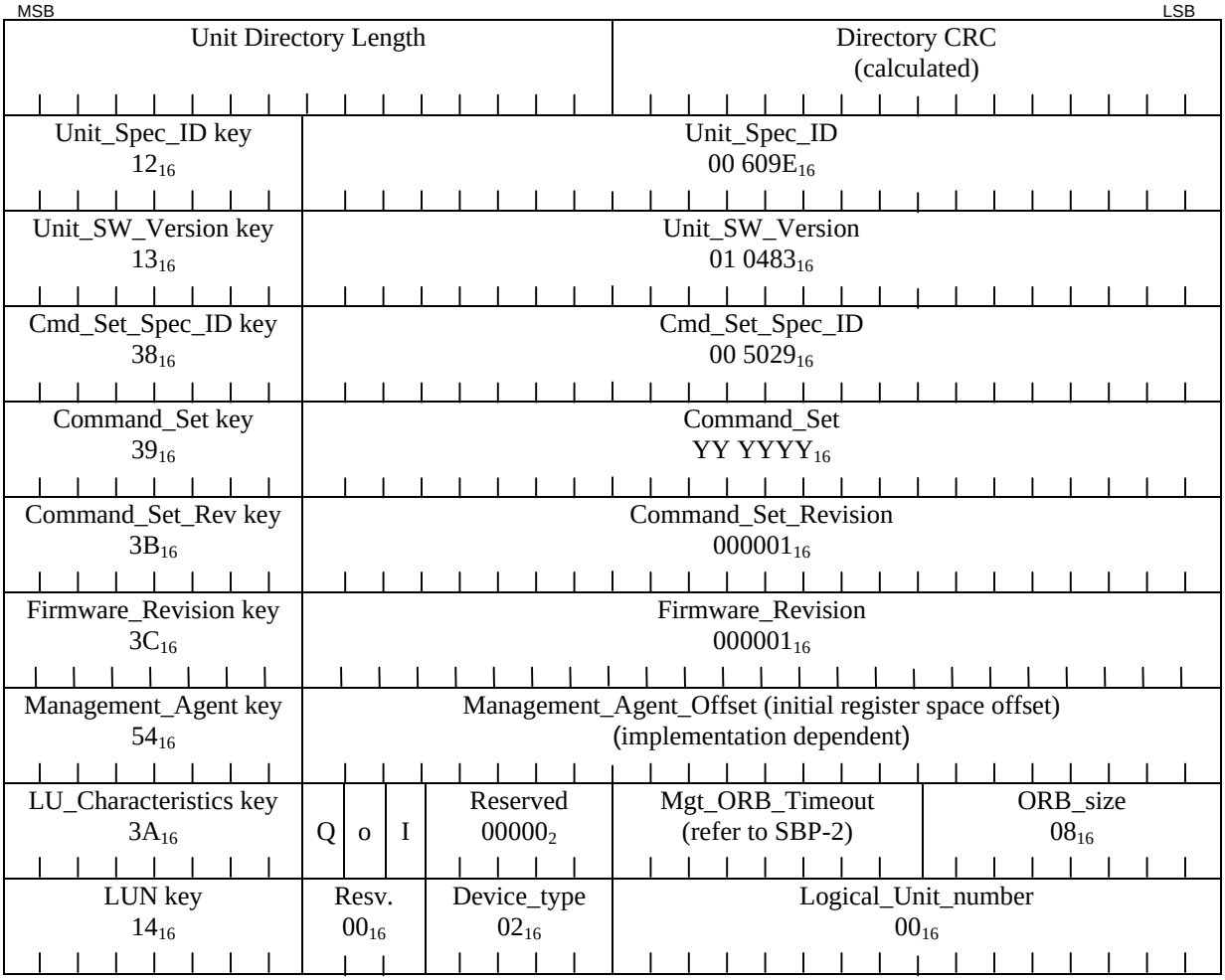
The second quadlet of the feature directory contains the Spec_ID for the directory. The Spec_ID value should contain the 24 bit OUI of the PWG.

The third quadlet of the feature directory contains the SW_Version for the directory. The SW_Version value shall be interpreted based on the Spec_ID value.

The remaining quadlets contain individual feature entries. The meaning of these feature entries is defined by the Spec_ID and SW_Version entries in this directory.

8.2.8 Unit Directory

Offset: 0450_{16}



Compliant devices shall implement at least one unit directory in the format specified by this profile. The unit directory shall contain Unit_Spec_ID and Unit_SW_Version entries as specified in ISO/IEC 13213:1994, a Management_Agent entry as specified by SBP-2, Cmd_Set_Spec_ID, Command_Set, Command_Set_Revision and Firmware_Revision entries as defined by this profile.

The first quadlet of the unit directory contains the directory_length and CRC-16 values. Each of these values is 16 bits in length.

The second quadlet contains the Unit_Spec_ID entry. The concatenated values of key type and key value for the Unit_Spec_ID is 12_{16}. The SBP-2 Unit_Spec_ID value is $00\ 609E_{16}$.

The third quadlet contains the Unit_SW_Version entry. The concatenated values of key type and key value for the Unit_SW_Version is 13_{16}. The SBP-2 Unit_SW_Version value is $01\ 0483_{16}$.

The fourth quadlet contains the Cmd_Set_Spec_ID entry. The concatenated values of key type and key value for the Cmd_Set_Spec_ID is 38_{16}. The 1394 PWG Profile Cmd_Set_Spec_ID value is 005029_{16}.

The fifth quadlet contains the Command_Set entry. The concatenated values of key type and key value for the Command_Set is 39_{16}. The 1394 PWG Profile Command_Set value is YY $YYYY_{16}$.

The sixth quadlet contains the Command_Set_Revision entry. The concatenated values of key type and key value for the Command_Set_Revision is $3B_{16}$. The 1394 PWG Profile Command_Set_Revision value is 00 0001_{16}.

The seventh quadlet contains the Firmware_Revision entry. The concatenated values of key type and key value for the Firmware_Revision is $3C_{16}$. The 1394 PWG Profile Firmware_Revision value is 00 0001_{16}.

The eighth quadlet contains the Management_Agent_Offset entry. The concatenated values of key type and key value for the Management_Agent_Offset is 54_{16}. The Management_Agent_Offset value is implementation dependent.

The ninth quadlet contains the LU_Characteristics entry. The concatenated values of key type and key value for the LU_Characteristics is $3A_{16}$. The 1394 PWG Profile LU_Characteristics value is $00A008_{16}$.

- The queuing model is defined by this profile and associated command set.
- The unordered execution model is supported.
- Asynchronous mode is used.
- Management ORB timeouts (refer to SBP-2 specification)
- The ORB size field is set to eight bytes.

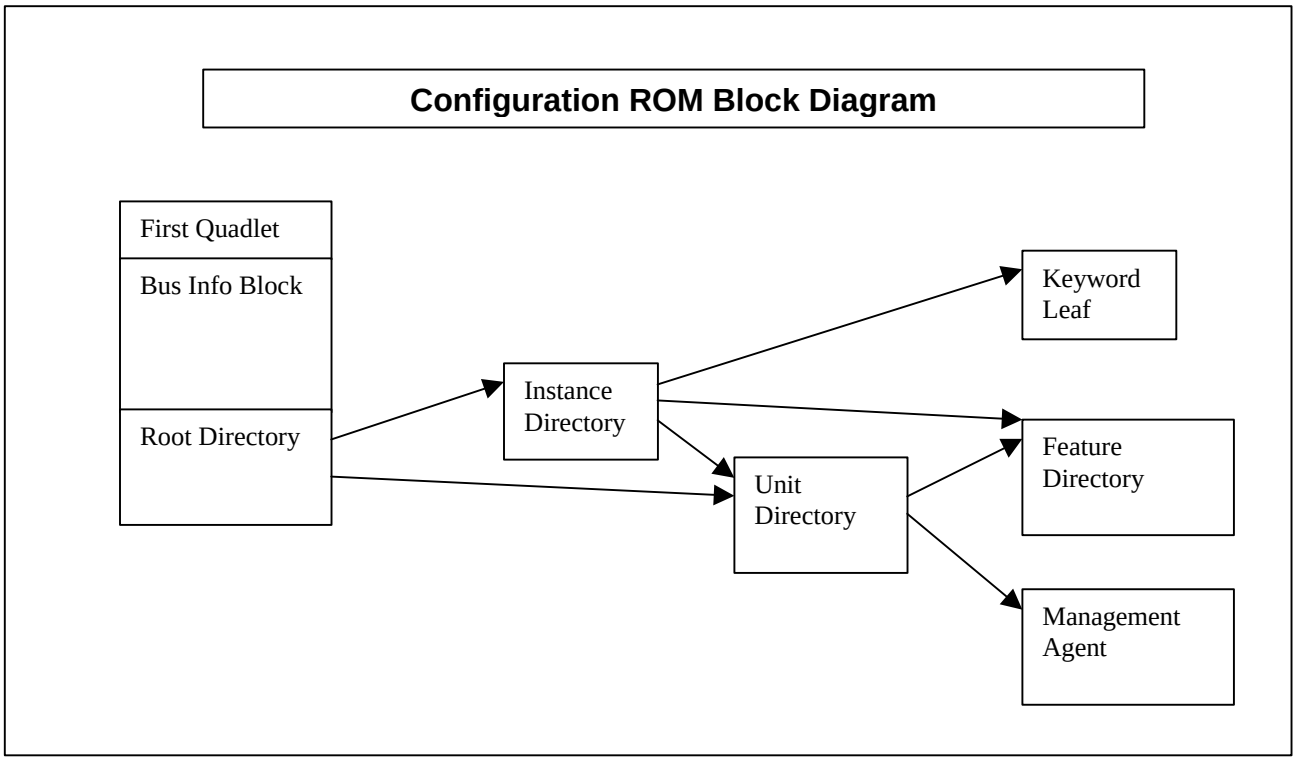
The tenth quadlet contains the Logical_Unit_Number entry. The concatenated values of key type and key value for the Logical_Unit_Number is 14_{16}. The value is divided into three fields: a group of reserved bits, a five bit device_type field, and a 16 bit Logical_Unit_Number field. Valid device_type values range from 00_{16} to $1F_{16.}$ Defined values are:

02_{16} - Printer
03_{16} - Processor
06_{16} - Scanner
09_{16} - Comm Device
$1F_{16}$ - Unknown – Needs Command_Set specific detection

8.3 Simple Imaging Device Example (Informative?)

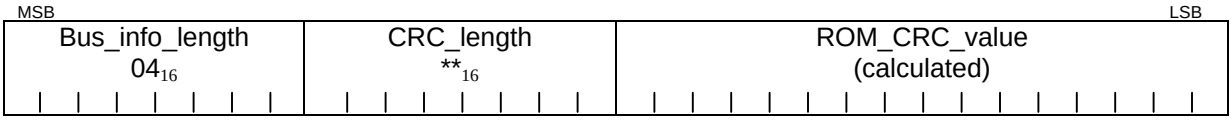
The sections contain information on a minimal implementation of Configuration ROM for a simple imaging device.

8.3.1 Block Diagram



8.3.2 First Quadlet

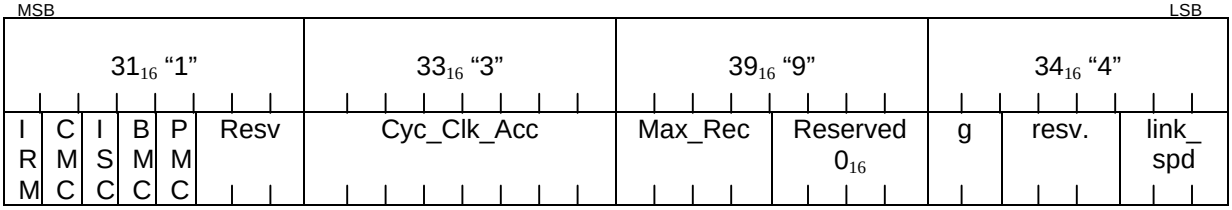
Offset: 0400_{16}



The first quadlet of configuration ROM is at the base address of FFFF F000 0400_{16}. A read of this location will indicate when the node completes initialization.

8.3.3 Bus_Info_Block

Offset: 0404_{16}



Node_Vendor_ID	Chip_ID_High
Chip_ID_Low	

8.3.4 Root Directory

Offset: 0414_{16}

MSB	LSB
Directory_length 05_{16}	CRC-16 (calculated)
Vendor ID key 03_{16}	Module_Vendor_ID (can be the same as Node_Vendor_ID)
Textual Descriptor key 81_{16}	Module_Vendor_ID textual descriptor offset (indirect offset) 0x????
Node Capabilities key $0C_{16}$	Node_Capabilities $0083C0_{16}$
Instance Directory key $D8_{16}$	Instance Directory offset (indirect offset)
Unit Directory key $D1_{16}$	Unit Directory offset (indirect offset)

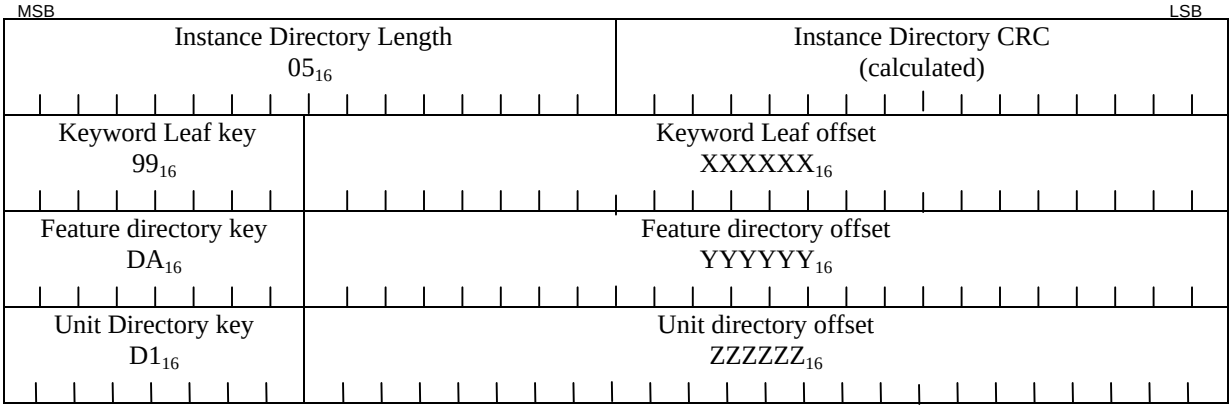
8.3.5 Vendor Textual Descriptor

Offset: $042C_{16}$

MSB			LSB
Leaf Length 05_{16}		Leaf CRC (calculated)	
Spec_type 00_{16}	Specifier_ID $00\ 0000_{16}$		
Language_ID $0000\ 0000_{16}$			
50_{16} “P”	72_{16} “r”	69_{16} “i”	$6E_{16}$ “n”
74_{16} “t”	65_{16} “e”	72_{16} “r”	20_{16} “ ”
43_{16} “C”	$6F_{16}$ “o”	2E “.”	00_{16}

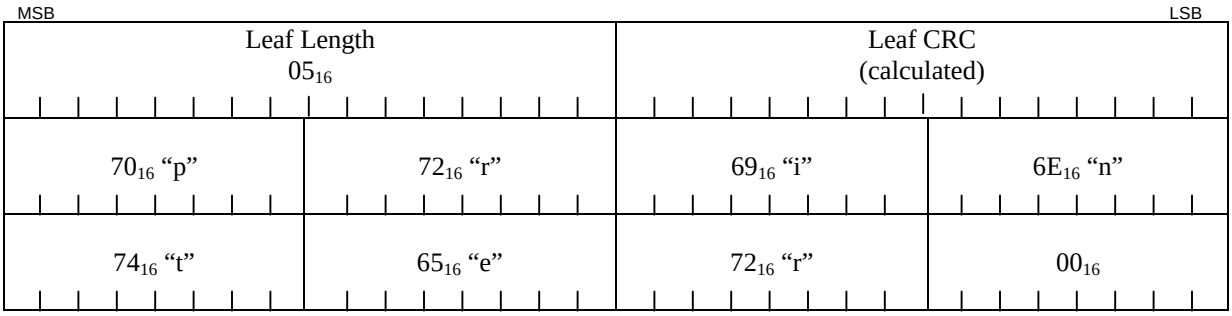
8.3.6 Instance Directory

Offset: 0444_{16}



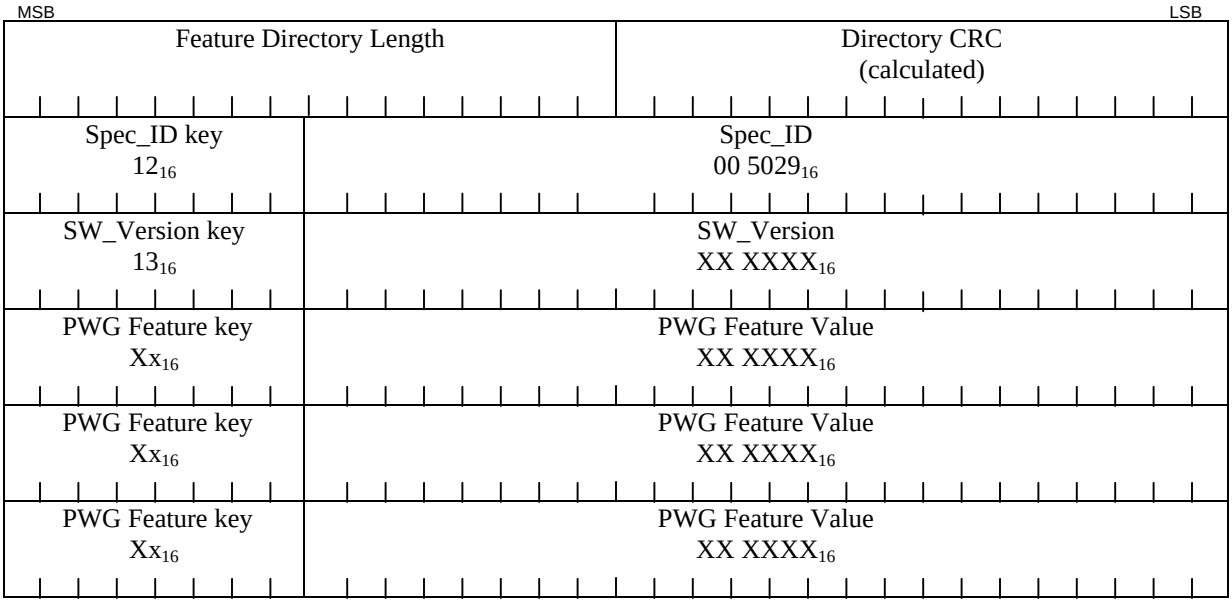
8.3.7 Keyword Leaf

Offset: 0444_{16}



8.3.8 Feature Directory

Offset: 0444_{16}



PWG Feature key Xx_{16}	PWG Feature Value XX $XXXX_{16}$

8.3.9 Unit Directory

Offset: 0450_{16}

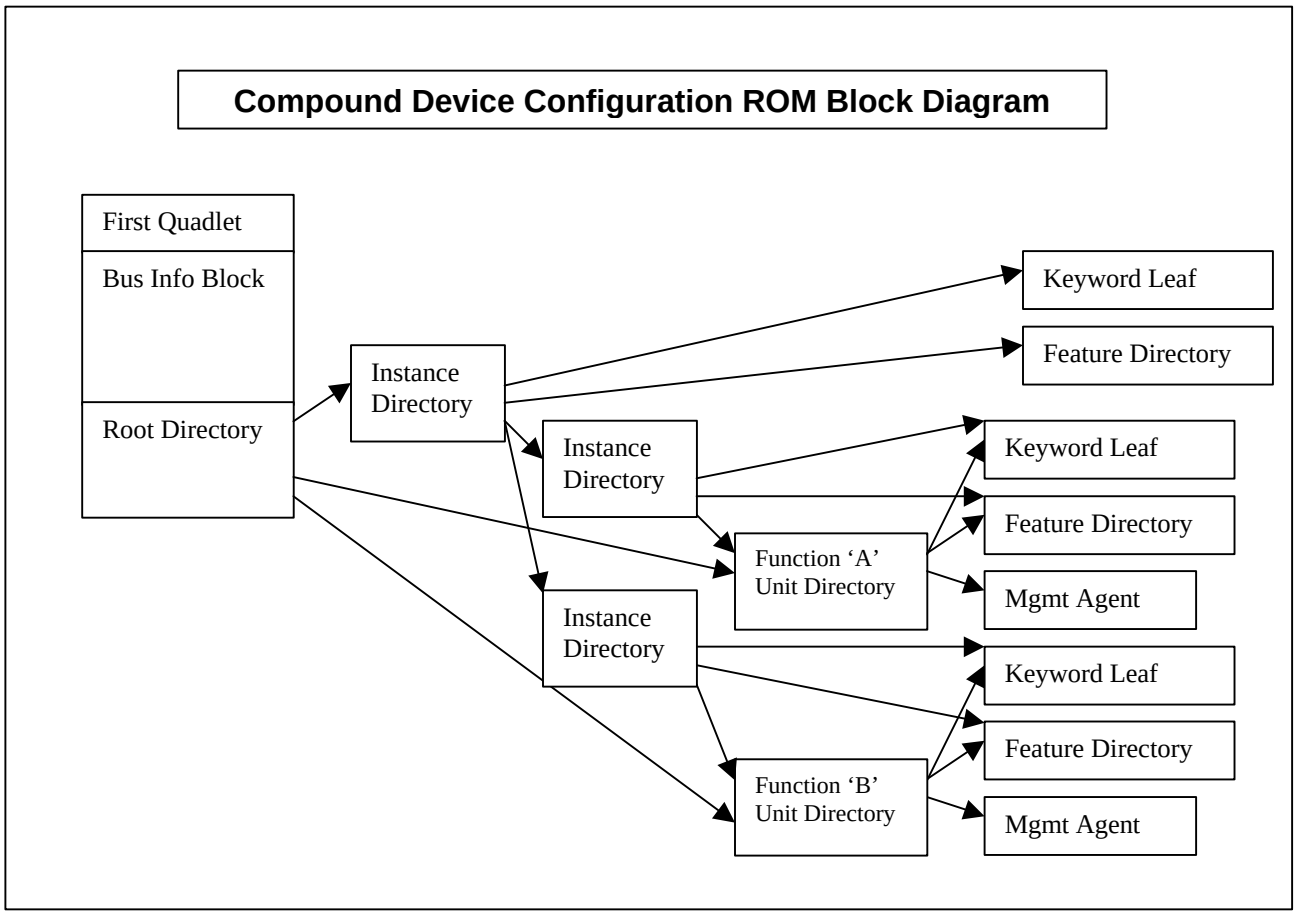
MSB LSB

Unit Directory Length					Directory CRC (calculated)	
Unit_Spec_ID key 12_{16}	Unit_Spec_ID 00 $609E_{16}$					
Unit_SW_Version key 13_{16}	Unit_SW_Version 01 0483_{16}					
Cmd_Set_Spec_ID key 38_{16}	Cmd_Set_Spec_ID 00 5029_{16}					
Command_Set key 39_{16}	Command_Set YY $YYYY_{16}$					
Command_Set_Rev key $3B_{16}$	Command_Set_Revision 000001_{16}					
Firmware_Revision key $3C_{16}$	Firmware_Revision 000001_{16}					
Management_Agent key 54_{16}	Management_Agent_Offset (initial register space offset) (implementation dependent)					
LU_Characteristics key $3A_{16}$	Q	o	I	Reserved 00000_2	Mgt_ORB_Timeout (refer to SBP-2)	ORB_size 08_{16}
LUN key 14_{16}	Resv. 00_{16}			Device_type 02_{16}	Logical_Unit_number 00_{16}	

8.4 Compound Imaging Device (Informative?)

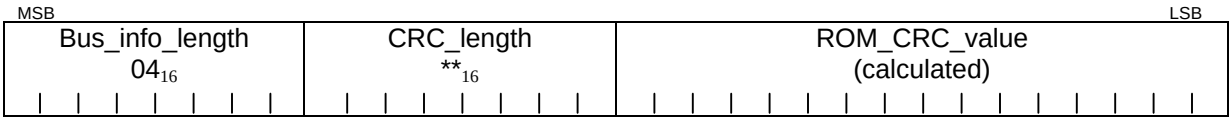
The sections contain information on a minimal implementation of Configuration ROM for a compound imaging device with two distinct functions. In this example, one function is a printer and the other is a scanner.

8.4.1 Block Diagram



8.4.2 First Quadlet

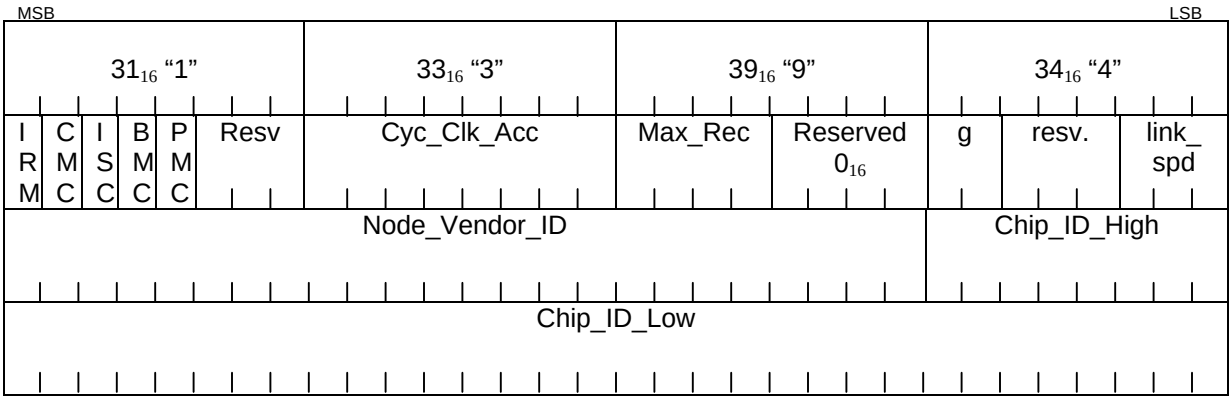
Offset: 0400_{16}



The first quadlet of configuration ROM is at the base address of FFFF F000 0400_{16}. A read of this location will indicate when the node completes initialization.

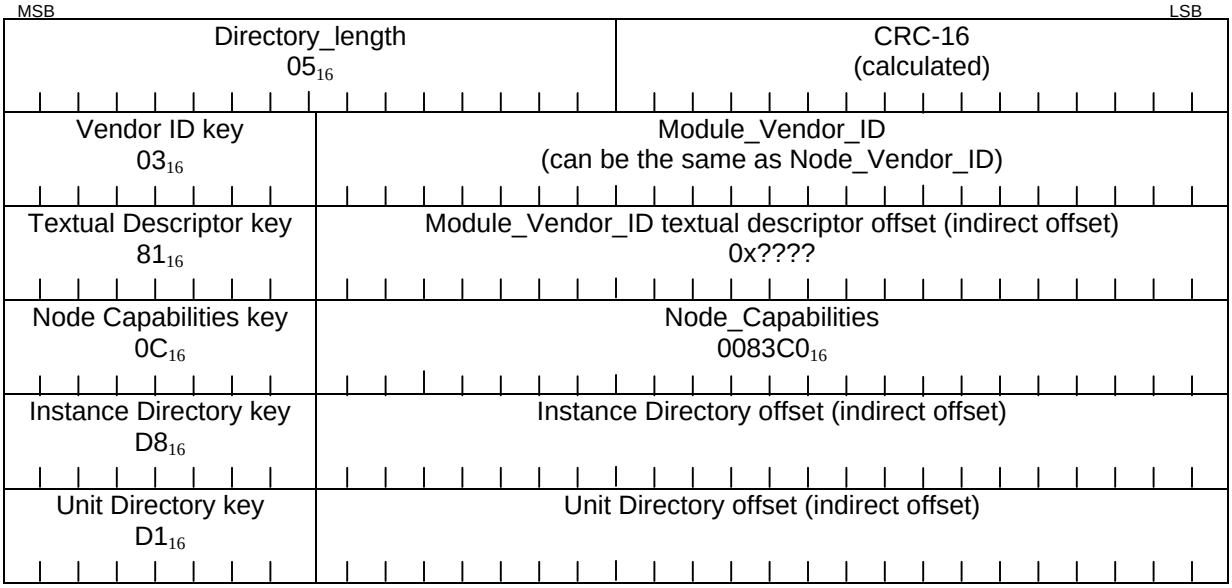
8.4.3 Bus_Info_Block

Offset: 0404_{16}



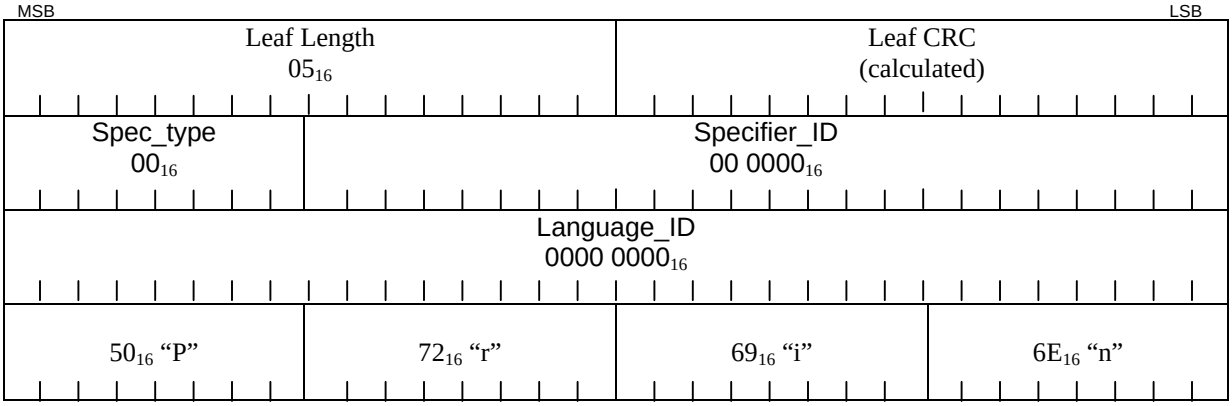
8.4.4 Root Directory

Offset: 0414_{16}



8.4.5 Vendor Textual Descriptor

Offset: $042C_{16}$



74_{16} “t”	65_{16} “e”	72_{16} “r”	20_{16} “ ”
43_{16} “C”	$6F_{16}$ “o”	2E “.”	00_{16}

8.4.6 Instance Directory (Root)

Offset: 0444_{16}

MSB LSB

Instance Directory Length 05_{16}	Instance Directory CRC (calculated)
Keyword Leaf key 99_{16}	Keyword Leaf offset $XXXXXX_{16}$
Feature directory key DA_{16}	Feature directory offset $YYYYYY_{16}$
Instance Directory key $D8_{16}$	Printer Instance directory offset $ZZZZZZ_{16}$
Instance Directory key $D8_{16}$	Scanner Instance directory offset $ZZZZZZ_{16}$

8.4.7 Keyword Leaf (Root)

Offset: 0444_{16}

MSB LSB

Leaf Length 05_{16}		Leaf CRC (calculated)	
70_{16} “p”	72_{16} “r”	69_{16} “i”	$6E_{16}$ “n”
74_{16} “t”	65_{16} “e”	72_{16} “r”	00_{16}

8.4.8 Feature Directory (Root)

Offset: 0444_{16}

MSB LSB

Feature Directory Length	Directory CRC (calculated)
Spec_ID key 12_{16}	Spec_ID $00\ 5029_{16}$

SW_Version key 13_{16}	SW_Version 01 0483_{16}
PWG Feature key Xx_{16}	PWG Feature Value XX $XXXX_{16}$

8.4.9 Instance Directory (Printer)

Offset: 0444_{16}

MSB LSB

Instance Directory Length 05_{16}	Instance Directory CRC (calculated)
Keyword Leaf key 99_{16}	Keyword Leaf offset $XXXXXX_{16}$
Feature directory key DA_{16}	Feature directory offset $YYYYYY_{16}$
Unit Directory key $D1_{16}$	Unit directory offset $ZZZZZZ_{16}$

8.4.10 Keyword Leaf (Root)

Offset: 0444_{16}

MSB LSB

Leaf Length 05_{16}		Leaf CRC (calculated)	
70_{16} “p”	72_{16} “r”	69_{16} “i”	$6E_{16}$ “n”
74_{16} “t”	65_{16} “e”	72_{16} “r”	00_{16}

8.4.11 Feature Directory (Root)

Offset: 0444_{16}

MSB LSB

Feature Directory Length	Directory CRC (calculated)
Spec_ID key 12_{16}	Spec_ID 00 5029_{16}
SW_Version key 13_{16}	SW_Version 01 0483_{16}

PWG Feature key Xx_{16}	PWG Feature Value XX $XXXX_{16}$

8.4.12 Instance Directory (Scanner)

Offset: 0444_{16}

MSB LSB

Instance Directory Length 05_{16}	Instance Directory CRC (calculated)
Keyword Leaf key 99_{16}	Keyword Leaf offset $XXXXXX_{16}$
Feature directory key DA_{16}	Feature directory offset $YYYYYY_{16}$
Unit Directory key $D1_{16}$	Unit directory offset $ZZZZZZ_{16}$

8.4.13 Keyword Leaf (Scanner)

Offset: 0444_{16}

MSB LSB

Leaf Length 05_{16}		Leaf CRC (calculated)	
70_{16} “p”	72_{16} “r”	69_{16} “i”	$6E_{16}$ “n”
74_{16} “t”	65_{16} “e”	72_{16} “r”	00_{16}

8.4.14 Feature Directory (Scanner)

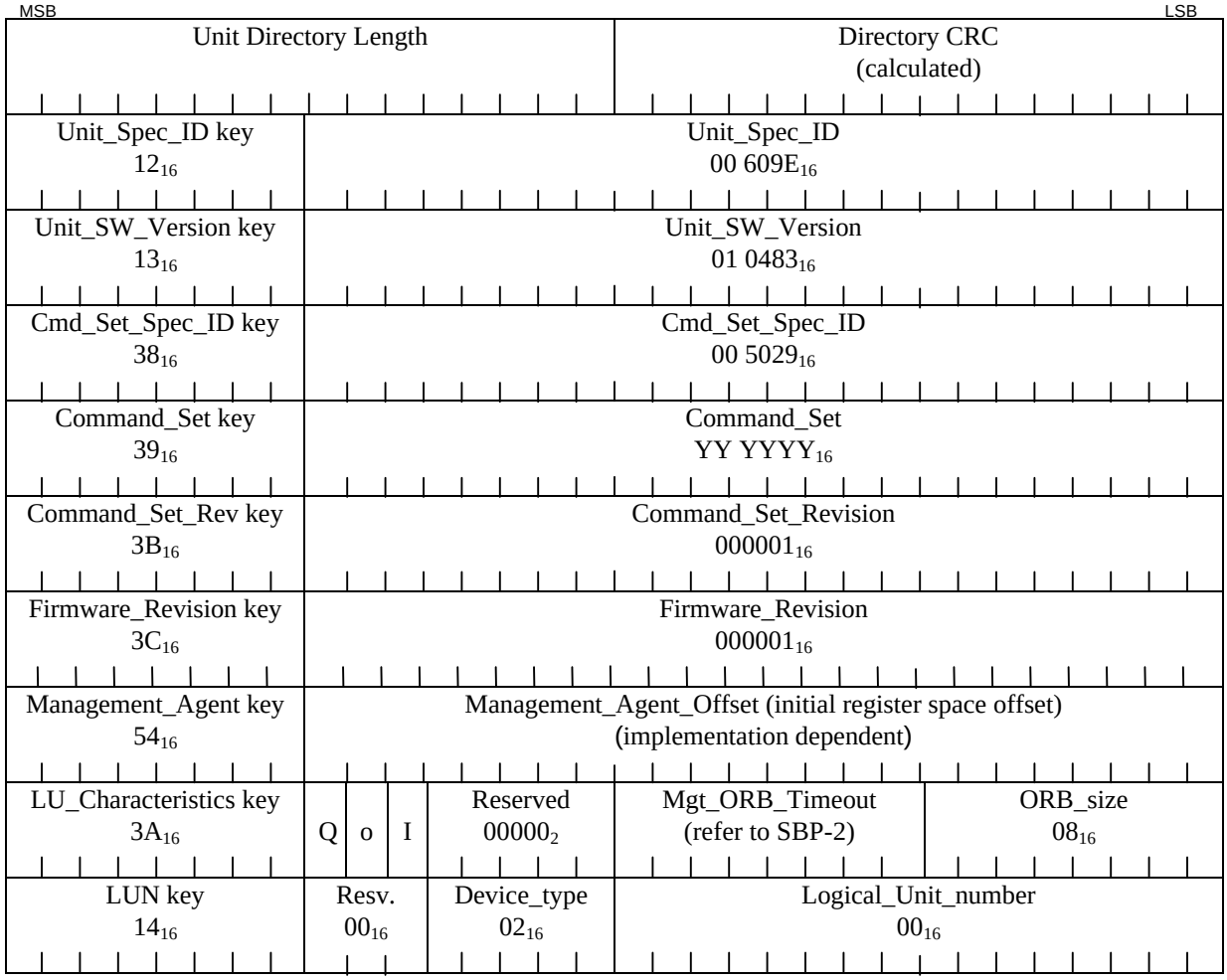
Offset: 0444_{16}

MSB LSB

Feature Directory Length	Directory CRC (calculated)
Spec_ID key 12_{16}	Spec_ID $00\ 5029_{16}$
SW_Version key 13_{16}	SW_Version $01\ 0483_{16}$
PWG Feature key Xx_{16}	PWG Feature Value XX $XXXX_{16}$

8.4.15 Unit Directory

Offset: 0450_{16}



9 Discovery (Informative)

The primary method for discovering devices on the Serial Bus is through information read from the Configuration ROM. This profile defines information in addition to that defined in the referenced specifications.

9.1 Device Information Model – Target

9.1.1 Availability

Availability of the configuration ROM data is determined by the first quadlet at location FFFF F000 $0400_{16.}$

9.1.2 Changes

Devices that change values in their configuration ROM shall only change those values during a bus reset.

9.1.3 Indicator

Devices shall implement the generate bits defined in IEEE-p1394a. The value of this field is incremented if any portion of the configuration ROM has changed during a bus reset. The coverage of the Bus_Info_Block in the first quadlet via the CRC_Length field causes the the CRC value in the first quadlet to be recalculated each time the generate bits are modified.

9.2 Device Information Model – Initiator

9.2.1 Device Availability

This section is provided to how a Serial Bus node can detect the availability of the compliant device configuration ROM.

Compliant hosts will read the first quadlet of configuration ROM at the base address of FFFF F000 0400_{16}. The configuration ROM of the target becomes available when the value read from this location is non-zero.

9.2.2 Device Class Detection

This section is provided to understand the detection mechanism for the device class.

1394 PWG Profile compliant nodes are required to implement an instance directory which contains a Keyword Leaf, Feature Directory and Unit_Directory_offset entries.

A bus enumerator can perform a top level search of the bus by inspecting the keyword leaf. The Feature Directory provides more specific information related to the device class listed in the Keyword Leaf.

In addition to these new extensions to Configuration ROM, legacy enumerators can inspect the five-bit device_type field of the Logical_Unit_Number entry in the Unit Directory.

9.2.3 Protocol Detection

This section is provided to understand the detection mechanism for the protocol driver stack.

SBP-2 compliant nodes are required to implement a unit directory that contains a Unit_Spec_ID and Unit_SW_Version entries. The concatenated values of key type and key value for the Unit_Spec_ID is 12_{16} and Unit_SW_Version is 13_{16}. The SBP-2 Unit_Spec_ID value is 00 $609E_{16}$ and the Unit_SW_Version value is 01 0483_{16}.

1394 PWG Profile compliant nodes are required to implement a unit directory which contains a Cmd_Set_Spec_ID and Command_Set entries. The concatenated values of key type and key value for the Cmd_Set_Spec_ID is 12_{16} and Unit_SW_Version is 13_{16}. The 1394 PWG Profile Cmd_Set_Spec_ID value is 005029_{16} and the Command_Set value is XX $XXXX_{16}$. In addition, the Logical_Unit_Number entry in the Unit Directory contains a five-bit device_type field.

9.2.4 Plug & Play Support

Devices may provide additional configuration ROM entries in addition to those defined in this profile. The specification for these additional entries is vendor dependent.

10 Identifiers

1394 nodes require 24 bit identifiers to correctly identify the software interface for a node.

10.1 SBP-2 Defined Identifiers

Unit_Spec_ID == 00 $609E_{16}$
Unit_SW_Version == 01 0483_{16}.

10.2 1394 PWG Organization Unique Identifier (OUI)

The 1394 PWG Profile implements the following references to 24 bit identifiers.

10.2.1 Cmd_Set_Spec_ID.

The 24 bit value for Cmd_Set_Spec_ID == XX $XXXX_{16}$

10.2.2 Command_Set.

The 1394 PWG Transport Command Set 24 bit value for Command_Set == 005029_{16}.

10.2.3 Feature Directory Spec_ID

Spec_ID == XX $XXXX_{16}$

10.3 1394 PWG Key Feature Directory Definitions

The 1394 PWG Profile defines the following keys.

10.3.1 Manufacturer_ID Entry

Manufacturer_ID == XX $XXXX_{16}$

10.3.2 Model_ID Entry

Manufacturer_ID == XX $XXXX_{16}$

10.3.3 Class_ID Entry

Manufacturer_ID == XX $XXXX_{16}$

10.3.4 Color_ID Entry

Manufacturer_ID == XX $XXXX_{16}$

10.3.5 Resolution_ID Entry

Manufacturer_ID == XX $XXXX_{16}$

10.4 OUI Source

OUI values are available from the IEEE Registration Authority Committee (RAC). Their address is:

Registration Authority Committee
The Institute of Electrical and Electronic Engineers, Inc.
445 Hoes Lane
Piscataway, NJ 08855-1331
USA
(908) 562 3813