iSCSI Boot Firmware Table (iBFT)

Note

Before using this information and the product it supports, read the information in “Notices”, on page 14.

1 iSCSI Boot Firmware Table (iBFT)

1.1 Introduction

The iSCSI Boot Firmware (iBF) Table (iBFT) is a block of information containing various parameters useful to the iSCSI Boot process. The iBFT is the mechanism by which iBF parameter values are conveyed to the operating system. The iBF builds and fills in the iBFT. The iBFT is available to the operating system to enable a consistent flow of the boot process.

1.2 Requirements

1. The various entries are compliant with the corresponding specification.
2. The table is compatible with an ACPI table format.
3. Support both IPV4 and IPV6 conventions.
4. Support various firmware packaging types:
    a. System ROM
    b. Adapter ROM
    c. Network Boot Program (NBP)
5. The iBFT must remain as compact as possible because it can be located in low memory which is a scarce resource. For example, in an NBP implementation it is possible that between 512k and 640k there is the EBDA, UNDI Stack, UNDI, PXE, and the NBP itself.

1.3 Conventions

1.3.1 IP Address

An example IPV4 address (192.168.70.50) stored in an IPV6 field:

IPv4-mapped IPv6 address

```
UCHAR exampleIP[16] = { 0x00, 0x00, 0x00, 0x00,
                        0x00, 0x00, 0x00, 0x00,
                        0x00, 0x00, 0xff, 0xff,
                        0xc0, 0xA8, 0x46, 0x23 };
```

An IP Address that is not present or not specified shall be all zeros.

See [Addr-Arch], [ipv4/6].

1.3.2 Alignment

Bytes (8-bit entries) are aligned on a byte boundary.

Words (16-bit entries) are aligned on an even byte boundary.

1.3.3 Endianess

All 2-byte entries are little endian "Words" (16-bit quantities) unless otherwise noted.

All 4-byte entries are little endian “DWords” (32-bit quantities) unless otherwise noted.

1.3.4 Length and Offset Values

Length	Offset	Description
0	0	Entry does not exist
0	Non-zero	Empty entry exists

1.3.5 Heap Array Format

A Heap Array entry is a collection of bytes. The total count of bytes of an entry is reflected in the corresponding Length field.

All array items stored in the Heap area will be followed by a separate NULL (a byte with a value of zero). This terminating NULL is not counted as part of the array length.

1.3.6 Parameter Origins

The iBF may obtain the necessary parameters using implementation specific methods. Some examples include: all information from DHCP, all information from local non-volatile storage (NVRAM), some information from DHCP and some from NVRAM, some information from network services such as iSNS, etc.

See [iscsi-boot].

1.3.7 Flag Bits

All Flag bits not explicitly identified are reserved and must be zero.

1.4 iBFT Layout

A compactness approach has been taken in defining the iBFT. Variable length fields are referenced by an offset mechanism.

1.4.1 iBFT Sections

The iBFT is built using a variety of sections.

Section	Byte Length	Byte Offset	Description
Header	48	0	Primary Header
Control	variable	48	Extended Header
Initiator	variable		Initiator Description
NIC	variable		NIC Description
Target	variable		Target Description
Heap	variable		Storage area for variable length values. String and blob Entries within the iBFT will point into this Heap area.

1.4.2 iBFT Standard Structure Header

Field	Byte Length	Byte Offset	Description
Structure ID	1	0	Structure ID
Version	1	1	Structure Version
Length	2	2	Structure Length
Index	1	4	Index
Flags	1	5	Structure Type Specific

1.4.3 iBF Table Header

Field	Byte Length	Byte Offset	Description
Signature	4	0	'iBFT' Signature for the iSCSI Boot Firmware Table
Length	4	4	Length in bytes of the entire IBFT, including the signature
Revision	1	8	Revision = 1
Checksum	1	9	Entire table must sum to zero
OEMID	6	10	OEM ID. All unused trailing bytes must be zero. [ACPI-OEMID]
OEM Table ID	8	16	For the iBFT the Table ID is the Manufacturer's Model ID. All unused trailing bytes must be zero.
Reserved	24	24	Reserved

1.4.3.1 Locating the iBFT

The iBFT is located by the following methods. A platform shall implement one or more of these methods.

1. The ACPI Method. The iBFT is pointed to by an entry in the RSDT/XSDT. Note that ACPI [ACPI=3.0b] specifies the string in the pointer as "IBFT" (all upper case) HOWEVER the signature in the table being pointed to is "iBFT" (note the mixed case).

2. The Low RAM Method. Scan for the table header signature in system memory between 512K and 1024K. The scan MUST be done starting at the lower address scanning forward to the higher address. When using the Low RAM Method the table header must be aligned on a 16-byte boundary.

Note: A system operating in UEFI mode shall utilize only the ACPI method.

1.4.4 Control Structure

Field	Byte Length	Byte Offset	Description
Structure ID	1	48 (0)	Structure ID = Control
Version	1	49 (1)	Structure Version = 1
Length	2	50 (2)	Structure Length >= 18
Index	1	52 (4)	Index = 0
Flags	1	53 (5)	Bit 0 : Target Login Mode Control 0 = Multi-Login Mode 1 = Single Login Mode
Structure Offsets			
Extensions	2	54 (6)	Optional. If unused must be zero. If used, must point to an Extensions Structure with a standard Structure header.
Initiator Offset	2	56 (8)	
NIC 0 Offset	2	58 (10)	
Target 0 Offset	2	60 (12)	
NIC 1 Offset	2	62 (14)	
Target 1 Offset	2	64 (16)	
Optional: Structure Exp.			

1.4.4.1 Structure Type / ID

Structure Type/ID:

0 = Reserved
1 = Control
2 = Initiator
3 = NIC
4 = Target
5 = Extensions

1.4.4.2 Control Structure Offsets

Unused Offsets shall be zero. For example, if NIC 1 and Target 1 are not used then the offset values shall be zero and no NIC 1 and Target 1 Structures are required to be present.

1.4.4.3 Optional Structure Expansion

The Control Structure can be expanded beyond the default minimum size. The Structure Length is used to compute the number of additional Optional Structure

Offsets. The type of the structure is determined by reading the corresponding Structure ID.

The Structures that are added, if any, do not need sequential Index values. The Index values may be sparse. For example, a single Structure Expansion entry may be added for NIC Index = 5.

1.4.4.4 Structure Alignment in Memory

Each Structure, if present, must be aligned on an 8 byte boundary.

1.4.4.5 Target Login Mode Control Flag

When this Flag is clear (0) the Initiator will attempt connection to all Targets specified in the Target Structures.

When this Flag is set (1) the Initiator will connect to only one Target. The Initiator will attempt to connect to a Target in the following order:

1. The Target indicated by the “Firmware Boot Selected Flag”.
2. The remaining Targets in ascending Index order.

1.4.5 Initiator Structure

Field	Byte Length	Byte Offset	Description
Structure ID	1	0	Structure ID = Initiator
Version	1	1	Structure Version = 1
Length	2	2	Structure Length = 74
Index	1	4	Index = 0
Flags	1	5	Bit 0 : Block Valid Flag 0 = no, 1=yes Bit 1 : Firmware Boot Selected Flag 0 = no, 1 = yes
iSNS Server	16	6	IP Address
SLP Server	16	22	IP Address
Primary Radius Server	16	38	IP Address
Secondary Radius Server	16	54	IP Address
Initiator Name Length	2	70	Heap Entry Length
Initiator Name Offset	2	72	Offset from the beginning of the iBFT

1.4.6 NIC Structure

Field	Byte Length	Byte Offset	Description
Structure ID	1	0	Structure ID = NIC
Version	1	1	Structure Version = 1
Length	2	2	Structure Length = 102

Index	1	4	Index = 0 for NIC 0
			Index = 1 for NIC 1
			…
			Index = n for NIC n

Flags	1	5	Bit 0 : Block Valid Flag 0 = no, 1=yes Bit 1 : Firmware Boot Selected Flag 0 = no, 1 = yes Bit 2 : Global / Link Local 0 = Link Local, 1 = Global
IP Address	16	6	IP Address
Subnet Mask Prefix	1	22	The mask prefix length. For example, 255.255.255.0 has a prefix length of 24
Origin	1	23	See [origin]
Gateway	16	24	IP Address
Primary DNS	16	40	IP Address
Secondary DNS	16	56	IP Address
DHCP	16	72	IP Address
VLAN	2	88	VLAN
MAC Address	6	90	MAC Address

PCI Bus/Dev/Func	2	96	Bus = 8 bits
			Device = 5 bits
			Function = 3 bits

Host Name Length	2	98	Heap Entry Length
Host Name Offset	2	100	Offset from the beginning of the iBFT In a DHCP scenario this can be the name stored as Option 12 host-name.

Host name if supplied must be in one of the following formats:
<host name>
<host name>.
<host name>.<domain>

1.4.7 Target Structure

Field	Byte Length	Byte Offset	Description
Structure ID	1	0	Structure ID = Target
Version	1	1	Structure Version = 1
Length	2	2	Structure Length = 54

Index	1	4	Index = 0 for Target 0
			Index = 1 for Target 1
			…
			Index = n for Target n

Flags	1	5	Bit 0 : Block Valid Flag 0 = no, 1=yes Bit 1 : Firmware Boot Selected Flag 0 = no, 1 = yes Bit 2 : Use Radius CHAP 0 = no, 1 = yes Bit 3 : Use Radius rCHAP 0 = no, 1 = yes
Target IP Address	16	6	IP Address
Target IP Socket	2	22	Likely 3260
Target Boot LUN	8	24	See [iscsi] Little Endian Quad Word
CHAP Type	1	32	0 = No CHAP 1 = CHAP 2 = Mutual CHAP
NIC Association	1	33	NIC Index
Target Name Length	2	34	Heap Entry Length
Target Name Offset	2	36	Offset from the beginning of the iBFT
CHAP Name Length	2	38	Heap Entry Length
CHAP Name Offset	2	40	Offset from the beginning of the iBFT
CHAP Secret Length	2	42	Heap Entry Length
CHAP Secret Offset	2	44	Offset from the beginning of the iBFT
Reverse CHAP Name Length	2	46	Heap Entry Length
Reverse CHAP Name Offset	2	48	Offset from the beginning of the iBFT
Reverse CHAP Secret Length	2	50	Heap Entry Length
Reverse CHAP Secret Offset	2	52	Offset from the beginning of the iBFT

CHAP – The Name/Password the Initiator sends to the Target.

Reverse CHAP – The Name/Password the Target sends to the Initiator.

1.5 References

[Addr-Arch] IP Version 6 Addressing Architecture
http://www.ietf.org/rfc/rfc2373.txt

[ipv4/6] See “2.1 Addresses”
http://www.ietf.org/rfc/rfc2765.txt

[iscsi] Internet Small Computer Systems Interface (iSCSI) RFC 3720
http://www.ietf.org/rfc/rfc3720.txt

[iscsi-boot] Bootstrapping Clients…
http://www.ietf.org/rfc/rfc4173.txt

[origin] Origin
http://msdn.microsoft.com/library/default.asp?url=/library/en-us/iphlp/iphlp/ip_prefix_origin.asp

[ACPI-3.0b] ACPI Specification Version 3.0b or later
http://www.acpi.info/

[ACPI-OEMID] A list of OEM IDs is maintained by the ACPI SIG. Contact the ACPI SIG Secretary for OEMID registration.

1.6 Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk. IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

(C) Copyright IBM Corp. 2004 439 Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
MW9A/050
5600 Cottle Road
San Jose, CA 95193
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurement may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

1.7 Trademarks

IBM the IBM logo, and BladeCenter are registered trademarks of IBM in the United States.

Other company, product, or service names may be trademarks or service marks of others.

End of Document