Optical Storage Technology Association

Document Change Notice 2-033

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Update of Identifier Suffix to 2.00*
Date: August 27, 1997

Description:

Since UDF Specification changes have been made the UDF Specification revision number stored in the Identifier Suffix of Entity Identifiers shall be updated to 2.00.

Change:

The section 2.1.5.3 on page 13 will be modified as follows:

2.1.5.3 IdentifierSuffix

The format of the *IdentifierSuffix* field is dependent on the type of the *Identifier*.

In regards to OSTA Domain *Entity Identifiers* specified in this document (appendix 6.1) the *IdentifierSuffix* field shall be constructed as follows:

Domain *IdentifierSuffix* field format

RBP	Length	Name	Contents
0	2	UDF Revision	Uint16 (= #0200)
2	1	Domain Flags	Uint8
3	5	Reserved	bytes (= #00)

The *UDFRevision* field shall contain **#0200** to indicate revision **2.00** of this document. This field will allow an implementation to detect changes made in newer revisions of this document.

In addition the following structure on page 13 will be modified as follows:

For implementation use *Entity Identifiers* defined by UDF (appendix 6.1) the *IdentifierSuffix* field shall be constructed as follows:

UDF *IdentifierSuffix*

RBP	Length	Name	Contents
0	2	UDF Revision	Uint16 (= #0200)
2	1	OS Class	Uint8
3	1	OS Identifier	Uint8
4	4	Reserved	bytes (= #00)

OSTA
Optical Storage
Technology Association

Document Change Notice
2-034

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Interpretation of Domain Flags*
Date: October 20, 1997

Description:

Guidelines for interpreting the Hard and Soft Write Protect flags

Add to section 3.3.6:

After the paragraph after the “Domain Flags” table:
The write protect flags appear in the Logical Volume Descriptor and in the File Set Descriptor. They shall be interpreted as follows:

```
is_fileset_write_protected = LVD.HardWriteProtect  || LVD.SoftWriteProtect ||
       FSD.HardWriteProtect  || FSD.SoftWriteProtect
is_fileset_hard_protected = LVD.HardWriteProtect   || FSD.HardWriteProtect
is_fileset_soft_protected = (LVD.SoftWriteProtect  || FSD.SoftWriteProtect) &&
       (! is_vol_hard_protected)
is_vol_write_protected = LVD.HardWriteProtect  || LVD.SoftWriteProtect
is_vol_hard_protected = LVD.HardWriteProtect
is_vol_soft_protected = LVD.SoftWriteProtect && !LVD.HardWriteProtect
```

Optical Storage Technology Association

Document Change Notice 2-035

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *OSTA Compressed Unicode updated to Unicode 2.0*
Date: August 27, 1997

Description:

OSTA Compressed Unicode will switch is basis from Unicode 1.1 to 2.0 to avoid incompatibilities with Unicode 1.1.

Change:

The section 2.1.1 Character Sets on page 8 will be modified as follows:

Replace:

"OSTA CS0 shall consist of the d-characters specified in the Unicode 1.1 standard (excluding #FEFF and FFFE) stored in the *OSTA Compressed Unicode* format which is defined as follows:"

With:

"OSTA CS0 shall consist of the d-characters specified in The Unicode Standard, Version 2.0 (ISBN 0-201-48345-9 from Addison-Wesley Publishing Company http://www.aw.com/devpress/, see also http://www.unicode.org/), excluding #FEFF and FFFE, stored in the *OSTA Compressed Unicode* format which is defined as follows:"

Search and replace "Unicode 1.1" with "Unicode 2.0" through out the document.

OSTA
Optical Storage
Technology Association

Document Change Notice
2-036

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Implementation of Multiple Data Stream Files*
Date: August 30, 1997

Description:

A new set of structures is defined for named streams of files. Named data stream support was originally developed for NT support, but is applicable across platforms.

Change:

Add section :

3.3.5 Named Streams

Named streams provide a mechanism for associating related data of a file. It is similar in concept to extended attributes. However, named streams have significant advantages over extended attributes. They are not as limited in length. Space management is much easier as each stream has its own space, rather than the common space of extended attributes. Finding a particular stream does not involve searching the entire data space, as it does for extended attributes.

Named streams are mainly intended for user data. For example, a database application may store the records in the default or main stream and indices in named streams. The user would then see only one file for the database rather than many, and the application can use the various streams almost as if they were independent files.

Named Streams are identified by an Extended File Entry. Extended File Entries are required for files with associated named streams. Files without named streams should use Extended File Entries. Files may have normal File Entries; normal File Entries would be used where backward compatibility is desired, such as writing DVD Video discs.

There is a “system stream directory” which is the stream directory identified by the File Set Descriptor. These streams are used to describe data related to the entire medium instead of data that relates to a file. UDF defines several “system streams” that are to be identified by this system stream directory.

It is recommended that Named Streams be used to store metadata and application data instead of Extended Attributes in new implementations.

3.3.5.1 Named Streams Restrictions

ECMA 167 revision 3 defines a new File Entry that contains a field for identifying a stream directory. This new File Entry should be used in place of the old File Entry, and should be used for describing the streams themselves. Old and new file entries may be freely mixed. In particular, compatibility with old reader implementations can be maintained for certain files.

Restrictions:

The stream directory ICB field of ICBs describing stream directories or named streams shall be set to zero. [no heirarchical streams]

Each named stream shall be identified by exactly one FID in exactly one Stream Directory. [no hard links among named streams or files and named streams]

Each Stream Directory ICB shall be identified by exactly one Stream Directory ICB field. [no hard links to stream directories]

Hard Links to files with named streams are allowed.

Named Streams and Stream Directories shall not have Extended Attributes.

The Unique ID field of Named Streams and Stream Directories shall be set to zero and shall be ignored when read. The Unique ID of a Named Stream or Stream Directory shall be considered the same as the Unique ID of the main data stream.

The UID, GID, and permissions fields of the main File Entry shall apply to all named streams associated with the main stream. At the time of creation of a named stream the values of the UID, GID and permissions fields of the main file entry should be used as the default values for the corresponding fields of the named stream. Implementations are not required to maintain or check these fields in a named stream.

Implementations should not present streams marked with the *metadata* bit set in the FID to the user. Streams marked with the *metadata* bit are intended solely for the use of the file system implementation.

The parent entry FID in a stream directory points to the main Extended File Entry, so its reference must be counted in the Link Count field of the Extended File Entry.

The modification time field of the main Extended File Entry should be updated whenever any associated named stream is modified. The Access Time field of the main Extended File Entry should be updated whenever any associated named stream is accessed. The SETUID and SETGID bits of the ICB Tag flags field in the main Extended File Entry should be cleared whenever any associated named stream is modified.

The ICB for a Named Stream directory shall have a file type of 13. All named streams shall have a file type of 5.

All systems shall make the main data stream available, even on implementations that do not implement named streams.

3.3.5.2 System Named Streams (Metadata)

A set of named streams is defined by UDF for file system use. Some UDF named streams are identified by the File Set Descriptor and apply to the entire file set (system stream directory). Others pertain to individual files or directories and are identified by the stream directory.

All UDF named streams shall have the Metadata bit set in the File Identifier Descriptor in the Stream Directory. All streams not generated by the file system implementation shall have this bit set to zero.

All UDF named streams shall have a file type of 5 in the ICB identifying the stream.

The four characters *UDF are the first four characters of all system named streams in this document. Implementations shall not use any identifier beginning with *UDF for metadata that is not defined in this document. All identifiers for metadata beginning with *UDF are reserved for future definition by OSTA.

3.3.5.3 Extended Attributes as named streams

An extended attribute may be recorded as a named stream instead. The extended attribute is converted according to the following rules:

The stream is marked as a Metadata stream.

The EA header and Header Checksum are not recorded. If the EA included pad bytes between the Header Checksum and the remaining data, these are also not recorded.

Any extended attribute of a file or directory can be converted to a stream of the same file or directory by the following algorithm:

1. Create a stream for the file or directory containing the extended attribute. The identifier specified for the Entity Identifier becomes the stream name.
2. Copy the data of the extended attribute into the stream.
3. Delete the extended attribute.

OSTA
Optical Storage
Technology Association

Document Change Notice
2-037

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *UDF Implementation Stream for Unique ID Mapping Data*
Date: October 7, 1997

Description:

Multiple platforms need the ability to locate the structures associated with a file or directory by an ID which uniquely identifies the name of a file or directory. A generalized solution to this problem supersedes the solution described in section 3.3.4.5.4.3 MacUniqueIDTable.

Change:

Insert a new section 2.3.4.2 and renumber existing 2.3.4.2 - 2.3.4.3.

2.3.4.2 struct long_ad ICB

The Implementation Use bytes of the long_ad in all File Identifier Descriptors shall be used to store the UDF Unique ID for the file and directory namespace.

UDF Unique ID

RBP	Length	Name	Contents
0	2	Reserved	bytes (= #00)
2	4	UDF Unique ID	Uint32

Section 3.2.1 Logical Volume Header Descriptor describes how UDF Unique ID field in Implementation Use bytes of the long_ad in the File Identifier Descriptor and the UniqueID file in the File Entry and Extended File Entry are set.

Replace Section 2.3.6.5 UniqueID with:

2.3.6.5 Uint64 UniqueID

For the root directory of a file set, this value shall be set to zero.

Section 3.2.1 Logical Volume Header Descriptor describes how the UDF Unique ID field in the Implementation Use bytes of the long_ad in the File Identifier Descriptor and the UniqueID file in the File Entry and Extended File Entry are set.

Replace Section 3.2.1.1 UniqueID with:

3.2.1.1 Uint64 UniqueID
This field contains the next *UniqueID* value which should be used. The field is initialized to 16, and it monotonically increases with each assignment described below. Whenever the lower 32-bits of this value reach #FFFFFFFF, the upper 32-bits is incremented by 1, as would be expected for a 64-bit value, but the lower 32-bits "wraps" to 16 (the initialization value). This behavior supports Mac™ OS which uses an ID number space of 16 through 2^32 - 1 inclusive, and will not cause problems for other platforms.

UniqueID is used whenever a new file or directory is created, or another name is linked to an existing file or directory. The File Identifier Descriptors and File Entries/Extended File Entries used for a stream directory and named streams associated with a file or directory do not use UniqueID; rather, the unique ID fields in these structures take their value from the UniqueID of the File Entry/Extended File Entry of the file/directory the streams are associated with. (Note, this is recursive in the case of hierarchical streams.)

When a file or directory is created, this UniqueID is assigned to the UniqueID field of the File Entry/Extended File Entry, the lower 32-bits of UniqueID are assigned to UDFUniqueID in the Implementation Use bytes of the long_ad in the File Identifier Descriptor (see 2.3.4.2), and UniqueID is incremented by the policy described above.

When a name is linked to an existing file or directory, the lower 32-bits of NextUniqueID are assigned to UDFUniqueID in the Implementation Use bytes of the long_ad in the File Identifier Descriptor (see 2.3.4.2), and UniqueID is incremented by the policy described above.

The lower 32-bits shall be the same in the File Entry/Extended File Entry and its first File Identifier Descriptor, but they shall differ in subsequent FIDs.

All UDF implementations shall maintain the UDFUniqueID in the FID and UniqueID in the FE/EFE as described in this section. The LVHD in a closed Logical Volume Integrity Descriptor shall have a valid UniqueID.

Section 3.3.4.5.4.3 MacUniqueIDTable is removed.

Add section 3.3.7.1 as follows:

3.3.7.1 Unique ID Mapping Data Stream

The Unique ID Mapping Data allows an implementation to go directly to the ICB hierarchy for the file/directory associated with a UDFUniqueID, or to the ICB hierarchy for the directory which contains the file/directory associated with the UDFUniqueID. Unique ID Mapping Data is stored as a named stream of the file described in 3.3.5. The name of this stream shall be set to:

"*UDF Unique ID Mapping Data"

The *Metadata* bit in the *File Characteristics* field of the File Identifier Descriptor shall be set to 1 to indicate that the existence of this file should not be made known to clients of a platform's file system interface.

shall be created for read-only media
shall be created by implementations which batch write (e.g., pre-mastering tools) a volume on write-once and rewritable media
for implementations which perform incremental updates of volumes on write-once or rewritable media (e.g., on-line file systems), the following rules apply:
may be created and maintained if not present
shall be maintained if present and volume is clean
should be repaired and maintained, but may be deleted, if present and volume is dirty
for these rules, a volume is clean if either a valid Close Logical Volume Integrity Descriptor or a valid Virtual Address Table is recorded

The contents of the stream is described by the table "UDF Unique ID Mapping Data" which contains some header fields before an array of "UDF Unique ID Mapping Entry." The fields of the structures are described below their corresponding table.

UDF Unique ID Mapping Entry

RBP	Length	Name	Contents
0	4	UDF Unique ID	Uint32
4	4	Parent Logical Block Number	Uint32
8	4	Object Logical Block Number	Uint32
12	2	Parent Partition Reference Number	Uint16
14	2	Object Partition Reference Number	Uint16

UDF Unique ID is the value found in a FID for the file or directory.

Parent Logical Block Number is the logical block number of the ICB identifying the directory that contains the FID identifying the object.

Object Logical Block Number is the logical block number of the ICB identifying this object.

Parent Partition Reference Number is the partition reference number from the long_ad of the ICB field in the parent in the same directory containing the FID for this file or directory.

Object Partition Reference Number is the partition reference number from the long_ad of the ICB field in the FID with this UDFUniqueID.

UDF Unique ID Mapping Data

RBP	Length	Name	Contents
0	32	Implementation Identifier	EntityID
32	4	Flags	Uint32
36	4	Mapping Entry Count (=MEC)	Uint32
40	8	Reserved	Bytes (= #00)
48	16*MEC	Mapping Entries	IDMappingEntry

Implementation Identifier is described in [cross reference to 2.1.5].

Flags are defined as follows:
Bit 0, If set to ONE, shall mean UDF Unique ID, once decremented by 16 (the value NextUniqueID is initialized to), can be used as an index into the array Mapping Entries. Blank entries, if present, are all beyond the last array element with a UDF Unique ID.
Bits 1 - 31, reserved, shall be set to ZERO.

Mapping Entry Count is the size, in entries, of the array Mapping Entries.

Mapping Entries is an array of UDF Unique ID Mapping Entry structures. There is one mapping entry for every non-stream, non-parent File Identifier Descriptor. Whenever the volume is consistent, the array is always sorted in ascending order of UDF Unique ID. Except as limited by the flags, blank entries are allowed anywhere in the array, and entries are not required to have a UDF Unique ID value of one more than the preceeding entry. A blank entry has a value of ZERO in all fields.

OSTA
Optical Storage
Technology Association

Document Change Notice
2-038

Document:	OSTA Universal Disk Format Specification Revision 1.50
Subject:	*Macintosh Resource Fork as a Named Stream*
Date:	October 7, 1997

Description:

A Name Stream will be used to store Mac™ OS Resource Forks.

Change:

Remove section 3.3.4.5.4.4 MacResourceFork

Add to the section system named streams:

Macintosh Resource Fork

The name of this stream shall be:

"*UDF Macintosh Resource Fork"

The *Metadata* bit in the *File Characteristics* field of the File Identifier Descriptor shall be set to 0 to indicate that the existence of this file may be made known to clients of a platform's file system interface.

OSTA
Optical Storage
Technology Association

Document Change Notice
2-039

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Termination of an extent of descriptors*
Date: February 12, 1998

Description:

ECMA 167 sections 3/8.4.2, 3/8.8.2 and 4/8.3.1 may be interpreted differently by different people. To eliminate any ambiguity, this DCN clarifies that a Volume Descriptor Sequence, Logical Volume Integrity Sequence and File Set Descriptor Sequence can all be terminated by reaching the end of the extent.

In Section 2. Basic Restrictions & Requirements:

In the entry for Volume Descriptor Sequence Extent, add this sentence at the end:
"The VDS Extent may be terminated by the extent length."

In the entry for Logical Volume Integrity Descriptor, add this sentence at the end:
"The Extent of LVIDs may be terminated by the extent length."

In the entry for File Set Descriptor, add this sentence at the end:
"The FSD Extent may be terminated by the extent length."

OSTA
Optical Storage
Technology Association

Document Change Notice
2-040

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Backwards Compatibility*
Date: October 7, 1997

Description:

The following change defines the requirements for UDF 2.00 implementations and backwards *read* compatibility to prior revisions of UDF formatted media.

Change:

The following will be added to section *1.2 Compliance* on page 3.

Backwards Read Compatibility – A compliant UDF 2.00 implementation *shall* be able to *read* all media written under UDF 1.50 and 1.02.
Backwards Write Compatibility – UDF 2.00 structures shall not be written to media which contains UDF 1.50 or UDF 1.02 structures. UDF 1.50 and UDF 1.02 structures shall not be written to media which contains UDF 2.00 structures. These two requirements prevent media from containing different versions of the UDF structures.

OSTA
Optical Storage
Technology Association

Document Change Notice
2-041

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Several changes to "Multisession and mixed mode"*
Date: February 12, 1998

Description:

This DCN adds several clarifications to section 6.10.3, "Multisession and Mixed Mode".

In section 6.10.3.1:

Change the last bullet from:

- "When recorded in Random Access mode, a duplicate Volume Recognition Sequence shall be recorded beginning at sector N-256."

to:

- "When recorded in Random Access mode, a duplicate Volume Recognition Sequence should be recorded beginning at sector N-16."

In section 6.10.3.3:

Add these two paragraphs:

"A new Main and Reserve Volume Descriptor Sequences may exist in each added session, and may be different from earlier VDSs."

"If the last session on a CD does not contain a valid UDF file system, the disc is not a UDF disc. Only the UDF structures in the last session, and any UDF structures and data referenced through them, are valid.

In section 6.10.3.3:

Change the sentence:

"The UDF session may contain pointers to data in other sessions, pointers to data only within the UDF session, or a combination of both."

to:

"The UDF session may contain pointers to data or metadata in other sessions, pointers to data or metadata only within the UDF session, or a combination of both."

OSTA
Optical Storage
Technology Association

Document Change Notice
2-042

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Note on recovery and Information Length*
Date: August 29, 1997

Description:

In some cases, the allocation descriptors cannot be used to reconstruct the file length.

Change:

Insert section 2.3.6.4:

2.3.6.4 Uint64 Information Length

In most cases, the Information Length can be reconstructed during a recovery operation by finding the sum of the lengths of each of the allocation descriptors. However, space may be allocated after the end of the file (identified as a "file tail.") As allocated and unrecorded space is a legal part of a file, using the allocation descriptors to determine information length will fail if the next to last allocation descriptor for the file identifies 2^30 - block size bytes, or if the next to last allocation descriptor is an integral multiple of the block size and the last allocation descriptor is not contiguous with the next to last allocation descriptor.

Please see ECMA 167 revision 3, part 4, section 12.1 for the schema for recording allocation descriptors.

OSTA
Optical Storage
Technology Association

Document Change Notice
2-043

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Location Field of Extended Attribute Header Descriptor*
Date: June 10, 1997

Description:

There is confusion about the value in the location fields of the Extended Attribute Header Descriptor.

Change:

Change UDF 3.3.4.1 Extended Attribute Header Descriptor

From:

"If the attributes associated with the *location* fields highlighted above do not exist, then the value of the *location* field shall point to the byte after the extended attribute space."

To:

- A value in one of the *location* fields highlighted above equal to or greater than the length of the EA space shall be interpreted as an indication that the corresponding attribute does not exist.

- If an attribute associated with one of the *location* fields highlighted above does not exist, then the value of the corresponding *location* field shall be set to #FFFFFFFF."

OSTA
Optical Storage
Technology Association

Document Change Notice
2-044

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Access Control Lists*
Date: August 30, 1997

Description:

Certain operating systems support the concept of Access Control Lists (ACLs) for enforcing file access restrictions. In order to facilitate support for ACL's UDF 2.0 will define a set of named streams, whose purpose is to store the ACL associated with a given file object.

Change:

ACLs under UDF will be stored as named streams, following the rules of section 3.3.5. The contents of a named stream ACL shall be opaque and are not defined by this document. Interpretation of the contents of a named stream ACL shall be left to the operating system for which the ACL is intended. The following names will be used to identify the ACLs and shall be reserved. These names shall not be used for application named streams.

"*UDF NT ACL"

This name shall identify the named stream ACL for theWindows NT operating system.

"*UDF UNIX ACL"

This name shall identify the named stream ACL for the UNIX operating system.

OSTA
Optical Storage
Technology Association

Document Change Notice
2-045

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Clarification of Vocabulary*
Date: August 27, 1997

Description:

Contributions to the specification have freely used new terms, rather than adopt the unambiguous vocabulary of ECMA 167. This DCN adds or changes definitions of UDF terminology in order to provide a clear mapping to the terms used in ECMA 167.

Change:

Modify/Add the following definitions in section 1.3.2:

1.3.2 Definitions

Logical Block Address	A logical block number [3/8.8.1]. **NOTE 1:** This is not to be confused with a logical block address [4/7.1], given by the the lb_addr structure which contains both a logical block number [3/8.8.1] and a partition reference number [3/8.8], the latter identifying the partition [3/8.7] which contains the addressed logical block [3/8.8.1]. **NOTE 2:** A logical block number [3/8.8.1] translates to a logical sector number [3/8.1.2] according to the scheme indicated by the partition map [3/10.7] of the partition [3/8.7] which contains the addressed logical block [3/8.8.1].
Media Block Address	A sector number [3/8.1.1], derived from the unique sector address given by a relevant standard for recording [1/5.10]. In this specification, a sector number [3/8.1.1] is equivalent to a a logical sector number [3/8.1.2].
Packet	A recordable unit, which is an integer number of contiguous sectors [1/5.9], which consist of user data sectors, and may include additional sectors [1/5.9] which are recorded as overhead of the Packet-writing operation and are addressable according to the relevant standard for recording [1/5.10].

Physical Address	A sector number [3/8.1.1], derived from the unique sector address given by a relevant standard for recording [1/5.10]. In this specification, a sector number [3/8.1.1] is equivalent to a a logical sector number [3/8.1.2].
Physical Block Address	A sector number [3/8.1.1], derived from the unique sector address given by a relevant standard for recording [1/5.10]. In this specification, a sector number [3/8.1.1] is equivalent to a a logical sector number [3/8.1.2].
physical sector	A sector [1/5.9] given by a relevant standard for recording [1/5.10]. In this specification, a sector [1/5.9] is equivalent to a a logical sector [3/8.1.2].
user data blocks	The logical blocks [3/8.8.1] which were recorded in the sectors [1/5.9] (equivalent in this specification to logical sectors [3/8.1.2]) of a Packet and which contain the data intentionally recorded by the user of the drive. This specifically does not include the logical blocks [3/8.8.1], if any, whose constituent sectors [1/5.9] were used for the overhead of recording the Packet, even though those sectors [1/5.9] are addressable according to the relevant standard for recording [1/5.10]. Like any logical blocks [3/8.8.1], user data blocks are identified by logical block numbers [3/8.8.1].
user data sectors	The sectors [1/5.9] of a Packet which contain the data intentionally recorded by the user of the drive, specifically not including those sectors [1/5.9] used for the overhead of recording the Packet, even though those sectors [1/5.9] may be addressable according to the relevant standard for recording [1/5.10]. Like any sectors [1/5.9], user data sectors are identified by sector numbers [3/8.1.1]. In this specification, a sector number [3/8.1.1] is equivalent to a a logical sector number [3/8.1.2].
Virtual Address	A logical block number [3/8.8.1] of a logical block [3/8.8.1] in a virtual partition. Such a logical block [3/8.8.1] is recorded using the space of a logical block [3/8.8.1] of a corresponding non-virtual partition. The Nth Uint32 in the VAT represents the logical block number [3/8.8.1] of a non-virtual partition used to record logical block number N of its corresponding virtual partition. The

first virtual address is 0.

virtual partition — A partition of a logical volume [3/8.8] identified in a logical volume descriptor [3/10.6] by a Type 2 partition map [3/10.7.3] recorded according section 2.2.8 of to this specification. The virtual partition map contains a partition number which is the same as the partition number [3/10.7.2.4] in a Type 1 partition map [3/10.7.2] in the same logical volume descriptor [3/10.6]. This partition number [3/10.7.2.4] identifies another, non-virtual, partition of the same logical volume [3/8.8]. Each logical block [3/8.8.1] in the virtual partition is recorded using the space of a logical block [3/8.8.1] of that corresponding non-virtual partition. A VAT lists the logical blocks [3/8.8.1] of the non-virtual partition which have been used used to record the logical blocks [3/8.8.1] of its corresponding virtual partition.

virtual sector — A logical block [3/8.8.1] in a virtual partition. Such a logical block [3/8.8.1] is recorded using the space of a logical block [3/8.8.1] of a corresponding non-virtual partition. A virtual sector should not be confused with a sector [1/5.9] or a logical sector [3/8.1.2].

VAT — A file [4/8.8] recorded in the space of a non-virtual partition which has a corresponding virtual partition, and whose data space [4/8.8.2] is structured according to section 2.2.10 of this specification. The first portion of this file provides an ordered list of Uint32s, where the Nth Uint32 represents the logical block number [3/8.8.1] of a non-virtual partition used to record logical block number N of its corresponding virtual partition. This file [4/8.8] is not necessarily referenced by a file identifier descriptor [4/14.4] of a directory [4/8.6] in the file set [4/8.5] of the logical volume [3/8.8].

OSTA
Optical Storage
Technology Association

Document Change Notice
2-046

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Miscellaneous Editorial Changes*
Date: October 7, 1997

Description:

.

Change:

ECMA 167 3rd Edition – Through out the entire UDF document all references to ISO 13346 will be changed to *ECMA 167 3rd Edition*. This change is required since because only ECMA 167 3rd Edition contains the modifications to support Streams.

Change:

Add the following note to section 3.3.4 Extended Attributes:

> NOTE: There may exist 2 Extended Attribute spaces per file, one embeded in the File Entry or Extended File Entry and the other as a separate space referenced by the Extended Attribute ICB address in the File Entry or Extended File Entry. Each Extended Attribute space, if present, must have its own Extended Attribute Header Descriptor (see the next section).

Change:

The section 6.9.3 Obtaining DVD Documents on page 106 will be modified as follows:

Replace:

To obtain a copy of the *DVD Specifications for Read-Only Disc* document as well as other DVD related material, contact:

> Toshiba Corporation
> Toshiba BLDG. 13D
> DVD Division
> 1-1 Shibaura 1-Chome, Minato-ku, Tokyo 105-01, JAPAN

Mr. Y. Mizutani
E-mail: 000092030295@tg-mail.toshiba.co.jp

With:
To obtain a copy of the *DVD Specifications for Read-Only Disc* document as well as other DVD related material, contact:

Toshiba Corporation
DVD Business Promotion & Support
DVD Products Division
Attn:Senior Manager
TEL :+81-3-3457-2473
FAX :+81-3-5444-9430

OSTA
Optical Storage
Technology Association

Document Change Notice
2-047

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Descriptor Tags of FIDs spanning block boundaries.*
Date: February 12, 1998

Description:

There is a problem when updating a directory on write once media: If a block is updated and rewritten at a new logical block, the Tag Location fields of the Descriptor Tags of all File Identifier Descriptors in that block need to be updated to reflect the new logical block number. If the Descriptor Tag of the last FID extends into the next block, that block will have to be rewritten to update the Tag Location field. This could conceivably cascade for several blocks, causing a minor update to require the rewriting of many blocks. If the Descriptor Tag field never spans a block boundary, the problem cannot occur.

This DCN restricts the Descriptor Tag fields from spanning block boundaries on write once media, to prevent this problem. Also, this DCN allows an optional optimization: The CRC length may be set to less than its full length. If the CRC doesn't extend into the next block, that block doesn't need to be read to recalculate the CRC.

Add to section 2.3.4.4:

When writing a File Identifier Descriptor to write-once media, to ensure that the Descriptor Tag field of the next FID will never span a block boundary, if there are less than 16 bytes remaining in the current block after the FID, the length of the FID shall be increased (using the Implementation Use field) enough to prevent this. The CRC length may be set to less than the size of the FID minus 16 (to not include the Implementation Use area).

OSTA
Optical Storage Technology Association

Document Change Notice
2-048

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Power Calibration Stream*
Date: October 7, 1997

Description:

This document modifies the proposed DCN that created the Power Calibration Extended Attribute such that it is a system stream instead of an Extended Attribute of the root directory.

Change:

The following shall be added to the section on system streams:

3.3.7.4 Power Calibration Stream

One of the potential limitations on the effective use of the packet-write capabilities of CD-Recordable drives is the limited number (100) of power calibration areas available on current CD-R media. These power calibration areas are used to establish the appropriate power calibration settings with which data can be successfully and reliably written to the CD-R disc currently in the drive. The appropriate settings for a specific drive can vary significantly from disc to disc, between two different drives of the same make and model, and even using the same disc, drive and system configuration, but under different environmental conditions.

Because of this, most current CD-R drives recalibrate themselves the first time a write is attempted after a media change has occurred. This imposes no restriction on recording to discs using the disc-at-once or track-at-once modes, since in each of these modes the disc will fill (either by consuming the total available data capacity or total number of recordable tracks) in less than 100 separate writes. When using packet-write though, the disc could be written to thousands of times over an extended period before the disc is full.

Suppose, for instance, one wanted to incrementally back-up any new and/or modified files at the end of each work day (though the drive might also be used intermittently to do other projects during the day). These back-ups may require writing as little as a megabyte (or even less) each day. If one of the power calibration areas is used to calibrate the drive before writing to the disc every day, within five months the power calibration areas will all have been used, but only a small fraction of the total disc

capacity will have been consumed. It is likely that such a result would be both unexpected and unacceptable to the user of such a product.

The industry is attempting to provide ways to reduce the frequency with which the power calibration area of a CD-Recordable disc must be used. At least one current CD-R drive model tries to remember the power calibration values last used for recording data on each of a small number of recently encountered discs. Most CD-Recordable drives provide a mechanism for the host software to retrieve from the drive the most recent power calibration settings used by the drive to record data on the current disc, and to restore and use such information at some future time.

The Power Calibration Table described herein would be used to store on the disc the power calibration information thus obtained for future use by compatible implementations. The table consists of a header followed by a list of records containing power calibration settings which have been used by various drives and/or hosts, under various conditions, to record data on this disc, as well as other relevant information which may be used to determine which of the recorded calibration settings may be appropriate for use in a future situation. While every effort has been made to anticipate and include all necessary information to make effective use of the recorded power calibration information possible, it is up to the individual implementation to determine if, when and how such information will actually be used.

The Power Calibration Table shall be recorded as a system stream of the File Set Descriptor according to the rules of 3.3.5. The name of the stream shall be as follows:

"*UDF Power Cal Table

Implementations that do not support the Power Calibration Table shall not delete this stream. Further, any implementation which supports and/or uses the Power Calibration Table shall not delete or modify any records from such table which the implementation, through its use thereof, did not clearly and specifically obsolete or update.

The stream shall be formatted as follows:.

Power Calibration Table Stream

RBP	Length	Name	Contents
0	32	Implementation Identifier	EntityID [UDF 2.1.5]
32	4	Number of Records	Uint32 [1/7.1.5]
56	*	Power Calibration Table Records	bytes

Implementation Identifier:

See UDF section 2.1.5.

Number of Records:

Shall specify the number of records contained in the power calibration table

Power Calibration Table Records:

A series of power calibration table records for drives which have written to this disc. The length of this table is variable, but shall be a multiple of four bytes. Recording of data in any unstructured field shall be left-justified and padded on the right with #20 bytes.

Power Calibration Table Record Layout

RBP	Length	Name	Contents
0	2	Record Length	Uint16 [1/7.1.3]
2	2	Drive Unique Area Length [DUA_L]	Uint16 [1/7.1.3]
4	32	Vendor ID	bytes
36	16	Product ID	bytes
52	4	Firmware Revision Level	bytes
56	16	Serial Number/Device Unique ID	bytes
72	8	Host ID	bytes
80	12	Originating Time Stamp	Timestamp [1/7.3]
92	12	Updated Time Stamp	Timestamp [1/7.3]
104	2	Speed	Uint16 [1/7.1.3]
106	6	Power Calibration Values	bytes
112	[DUA_L]	Drive Unique Area	bytes

Record Length - The length of this Power Calibration Table Record in bytes, including the optional variable length Drive Unique Area. Shall be a multiple of four bytes.

Drive Unique Area Length - The length of the optional Drive Unique Area recorded at the end of this record in bytes. Shall be a multiple of four bytes.

Vendor ID - The Vendor ID reported by the drive.

Product ID - The Product ID reported by the drive.

Firmware Revision Level - The Firmware Revision Level reported by the drive.

Serial Number/Device Unique ID - A serial number or other unique identifier for the specific drive, of the model specified by the vendor and product IDs given, which has successfully used the power calibration values reported herein to record data on this disc.

Host ID - The host serial number, ethernet ID, or other value (or combination of values) used by an implementation to identify the specific host computer to which the drive was attached when it successfully used the power calibration values reported herein to record data on this disc. An implementation shall attempt to provide an unique value for each host, but is not required to guarantee the value's uniqueness.

Originating Time Stamp - The date and time at which the power calibration values recorded herein were initially verified to have been successully used.

Updated Time Stamp - The date and time at which the power calibration values recorded herein were most recently verified to have been successully used.

Speed - The recording speed, as reported by the drive, at which power calibration values recorded herein were successfully used. This value is the number of kilobytes per second (bytes per second / 1000) that the data was written to the disc by the drive (truncating any fractions). For example, a speed of 176 means data was written to the disc at 176 Kbytes/second, which is the basic CD-DA (Digital Audio) data rate (a.k.a. "1X" for CD-DA). A speed of 353 means data was written to the disc at 353 Kbytes/second, or twice the basic CD-DA data rate (a.k.a. "2X" for CD-DA). CD-ROM recording rates should be adjusted upward (roughly 15%) to the corresponding CD-DA rates to determine the correct speed value (eg. A "1X" CD-ROM data rate should be recorded as a "1X" CD-DA, which is a speed of 176). Note that these are raw data rates and do not refelect all overhead resulting from (additional) headers, error correction data, etc.

Power Calibration Values - The vendor-specific power calibration values reported by the drive.

Drive Unique Area - Optional area for recording unrestricted information unique to the drive (such as drive operating temperature) which certain implementations may use to enhance the use of the recorded power calibration information or the operation of the associated drive. Recording of data in this field shall be defined by the drive manufacturer. This area shall be an integral multiple of four bytes in length.

OSTA
Optical Storage Technology Association

Document Change Notice
2-049

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Update of File Set Descriptor for Streams*
Date: February 2, 1998

Description:

Update of File Set Descriptor to match ECMA 167 3rd edition.

Change section 2.3.2:

```
struct FileSetDescriptor {  /* ECMA 167 4/14.1 */
        struct tag          DescriptorTag;
        struct timestamp    RecordingDateandTime;
        Uint16              InterchangeLevel;
        Uint16              MaximumInterchangeLevel;
        Uint32              CharacterSetList;
        Uint32              MaximumCharacterSetList;
        Uint32              FileSetNumber;
        Uint32              FileSetDescriptorNumber;
        struct charspec     LogicalVolumeIdentifierCharacterSet;
        dstring             LogicalVolumeIdentifier[128];
        struct charspec     FileSetCharacterSet;
        dstring             FileSetIdentifer[32];
        dstring             CopyrightFileIdentifier[32];
        dstring             AbstractFileIdentifier[32];
        struct long_ad      RootDirectoryICB;
        struct EntityID     DomainIdentifier;
        struct long_ad      NextExtent;
        struct long_ad      StreamDirectoryICB;
        byte                Reserved[32];
}
```

Optical Storage Technology Association

Document Change Notice
2-050

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Support for CD-R Multisession required.*
Date: August 29, 1997

Description:

This document clarifies that support for CD-R Multisession is required.

Change:

Change **1.2 Compliance** to insert the following bullet after the *Media support* bullet:

- *Multisession support*. Any implementation that supports reading of CD-R media shall support reading of CD-R Multisessions as defined in 6.10.3.

Optical Storage Technology Association

Document Change Notice 2-051

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Value Fields in LVID for Virtual Partition on CD-R*
Date: August 27, 1997

Description:

This document defines the value of the FreeSpaceTable and SizeTable fields of the Logical Volume Integrity Descriptor for virtual partitions on CD-R media

Change:

Change sections **2.2.6.2 Uint32 FreeSpaceTable** and **2.2.6.3 Uint32 SizeTable** to:

2.2.6.2 Uint32 FreeSpaceTable Since most operating systems require that an implementation provide the true free space of a Logical Volume at mount time it is important that these values be maintained for all non-virtual partitions. The optional value of #FFFFFFFF, which indicates that the amount of available free space is not known, shall not be used for non-virtual partitions. For virtual partitions the FreeSpaceTable shall be set to #FFFFFFFF.
NOTE: The FreeSpaceTable is guaranteed to be correct only when the *Logical Volume Integrity Descriptor* is closed.

2.2.6.3 Uint32 SizeTable Since most operating systems require that an implementation provide the total size of a Logical Volume at mount time it is important that these values be maintained for all non-virtual partitions. The optional value of #FFFFFFFF, which indicates that the partition size is not known, shall not be used for non-virtual partitions. For virtual partitions the SizeTable shall be set to #FFFFFFFF.

OSTA
Optical Storage
Technology Association

Document Change Notice
2-052

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Change to Name Translation Algorithms*
Date: August 27, 1997

Description:

This document changes both of the name translation algorithms, eliminating the input parameters fidName and fidNameLen.

Change:

Change section **6.5 CRC Calculation** to add a new function, unicode_cksum:

```
unsigned short
unicode_cksum(s, n)
        register unsigned short *s;
        register int n;
{
        register unsigned short crc=0;
        while (n-- > 0) {
                /* Take high order byte first--corresponds to a big endian byte stream. */
                crc = crc_table[(crc>>8 ^ (*s>>8            ) & 0xff] ^ (crc<<8);
                crc = crc_table[(crc>>8 ^ (*s++ & 0xff)) & 0xff] ^ (crc<<8);
        }
        return crc;
}
```

Change:

Change section **6.7.1 DOS Algorithm**, to change the cksum function prototype (page 86) from:

```
unsigned short cksum(register unsigned char *s, register int n);
```

to:

```
unsigned short unicode_cksum(register unsigned short *s, register int n);
```

Change:

Change section **6.7.1 DOS Algorithm**, to change the parameter list (page 86) from:

```
int UDFDOSName(
```

```
unicode_t *dosName,          /* (Output)DOS compatible name. */
unicode_t *udfName,          /* (Input) Name from UDF volume. */
int udfLen,                  /* (Input) Length of UDF Name. */
byte *fidName,               /* (Input) Bytes as read from media */
int fidNameLen)              /* (Input) Number of bytes in fidName.*/
```

to:

```
int UDFDOSName(
unicode_t *dosName,          /* (Output)DOS compatible name. */
unicode_t *udfName,          /* (Input) Name from UDF volume. */
int udfLen)                  /* (Input) Length of UDF Name. */
```

Change:

Change section **6.7.1 DOS Algorithm**, to change the call to function cksum (page 88) from:

```
valueCRC = cksum (fidName, fidNameLen);
```

to:

```
valueCRC = unicode_cksum (udfName, udfLen);
```

Change:

Change section **6.7.2 OS/2, Macintosh, Windows 95 Windows NT and UNIX Algorithm**, to change the cksum function prototype (page 90) from:

```
unsigned short cksum(register unsigned char *s, register int n);
```

to:

```
unsigned short unicode_cksum(register unsigned short *s, register int n);
```

Change:

Change section **6.7.2 OS/2, Macintosh, Windows 95 Windows NT and UNIX Algorithm**, to change the parameter list (page 91) from:

```
int UDFDOSName(
unicode_t *dosName,          /* (Output)DOS compatible name. */
unicode_t *udfName,          /* (Input) Name from UDF volume. */
int udfLen,                  /* (Input) Length of UDF Name. */
byte *fidName,               /* (Input) Bytes as read from media */
int fidNameLen)              /* (Input) Number of bytes in fidName.*/
```

to:

```
int UDFDOSName(
unicode_t *dosName,          /* (Output)DOS compatible name. */
```

```
unicode_t *udfName,          /* (Input) Name from UDF volume. */
int udfLen)                  /* (Input) Length of UDF Name. */
```

Change:

Change section **6.7.2 OS/2, Macintosh, Windows 95 Windows NT and UNIX Algorithm**, to change the call to function cksum (page 92) from:
valueCRC = cksum (fidName, fidNameLen);

to:
valueCRC = unicode_cksum (udfName, udfLen);

Change:

The two name translation algorithms seem to be erroneously repeated at the end of the specification, after the Revision history, from page 113 (120 on the PDF reader) through page 125 (132 on the PDF reader). These pages should be deleted.

OSTA
Optical Storage
Technology Association

Document Change Notice
2-053

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Sparing Table Descriptor Tag*
Date: February 12, 1998

Description:

Section 2.2.11describes a Descriptor Tag with a Tag Identifier of 0; ECMA-167 3/7.2.1 says that the format of such a Descriptor Tag is not specified. This DCN clarifies the format of this Descriptor Tag.

In section 2.2.11:

Change:

This structure may be larger than a single sector if necesary.

- Descriptor Tag
  Contains 0, indicating that the contents are not specified by ECMA 167.

to:

This structure may be larger than a single sector if necesary.

- Descriptor Tag
  Contains a Tag Identifer of 0, which indicates that the format of the Descriptor Tag is not specified by ECMA 167. All other fields of the Descriptor Tag shall be valid, as if the Tag Identifier were one of the values defined by ECMA 167.

OSTA
Optical Storage
Technology Association

Document Change Notice
2-054

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Clarification of Logical Blocks Recorded Field*
Date: October 7, 1997

Description:

This DCN clarifies the value of the Logical Blocks recorded field in the File Entry for files with embedded data.

Change:

Add the following:

2.3.6.5 Uint64 Logical Blocks Recorded
For files and directories with embedded data the value of this field shall be ZERO.

OSTA
Optical Storage
Technology Association

Document Change Notice
2-055

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *System stream to indicate the time of last volume backup*
Date: February 12, 1998

Description:

This DCN defines a system stream to indicate the time of the last backup of the volume.

Add section 3.3.7.4:

3.3.7.4 UDF Backup Time

The name of this stream shall be set to:

"*UDF Backup"

This stream shall have the following contents, which should be embedded in the ICB:

UDF Backup TIme

RBP	Length	Name	Contents
0	12	Backup Time	timestamp

Backup Time is the latest time that a backup of this volume was performed.

OSTA
Optical Storage
Technology Association

Document Change Notice
2-056

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Replacement of VAT*
Date: September 29, 1997

Description:

New fields are required in the VAT to support operating systems that need a count of files and directories. This change describes a new VAT to be used on 2.0 media instead of the VAT described in UDF 1.5.

Change:

Replace section 2.2.10:

2.2.10 Virtual Allocation Table (VAT)

The Virtual Allocation Table (VAT) is used on sequentially written media (eg. CD-R) to give the appearance of randomly writable media to the system. The existence of this partition is identified in the partition maps. The VAT shall only be recorded on sequentially written media (eg. CD-R).

The VAT is a map that translates Virtual Addresses to logical addresses. It shall be recorded as a file identified by a File Entry ICB (VAT ICB) which allows great flexibility in building the table. The VAT ICB is the last sector recorded in any transaction. The VAT itself may be recorded at any location.

The VAT shall be identified by a File Entry ICB with a file type of 248. This ICB shall be the last valid data sector recorded. Error recovery schemes can find the last valid VAT by finding ICBs with file type 248.

This file, when small, can be embedded in the ICB that describes it. If it is larger, it can be recorded in a sector or sectors preceding the ICB. The sectors do not have to be contiguous, which allows writing only new parts of the table if desired. This allows small incremental updates, even on disks with many directories.

When the VAT is small (a small number of directories on the disk), the VAT is updated by writing a new file ICB with the VAT embedded. When the VAT becomes too large to fit in the ICB, writing a single sector with the VAT and a second sector with the ICB is required. Beyond this point, more than one sector is required for the VAT. However, as multiple extents are supported, updating the VAT may consist of writing only the sector or sectors that need updating and writing the ICB with pointers to all of the pieces of the VAT.

The Virtual Allocation Table is used to redirect requests for certain information to the proper logical location. The indirection provided by this table provides the appearance of direct overwrite capability. For example, the ICB describing the root directory could be referenced as virtual sector 1. A virtual sector is contained in a partition identified by a virtual partition map entry. Over the course of updating the disk, the root directory may change. When it changes, a new sector describing the root directory is written, and its Logical Block Address is recorded as the Logical Block Address corresponding to virtual sector 1. Nothing that references virtual sector 1 needs to change, as it still points to the most current virtual sector 1 that exists, even though it exists at a new Logical Block Address.

The use of virtual addressing allows any desired structure to become effectively rewritable. The structure is rewritable when every pointer that references it does so only by its Virtual Address. When a replacement structure is written, the virtual reference does not need to change. The proper entry in the VAT is changed to reflect the new Logical Block Address of the corresponding Virtual Address and all virtual references then indirectly point to the new structure. All structures that require updating, such as directory ICBs, shall be referenced by a Virtual Address. As each structure is updated, its corresponding entry in the VAT ICB shall be updated.

The VAT shall be recorded as a sequence of Uint32 entries in a file. Each entry shall be the offset, in sectors, into the physical partition in which the VAT is located. The first entry shall be for the virtual partition sector 0, the second entry for virtual partition sector 1, etc. The Uint32 entries shall follow the VAT header.

The entry for the previous VAT ICB allows for viewing the file system as it appeared in an earlier state. If this field is #FFFFFFFF, then no such ICB is specified.

Virtual Allocation Table structure

Offset	Length	Name	Contents
0	2	Length of Header (=L_HD)	Uint16
2	2	Length of Implementation Use (=L_IU)	Uint16
4	128	Logical Volume Identifier	dstring
132	4	Previous VAT ICB location	Uint32
136	4	Number of FIDs identifying Files	Uint32
140	4	Number of non-parent FIDs identifying Directories	Uint32
144	2	Min UDF Read version	Uint16
146	2	Min UDF Write version	Uint16
148	2	Max UDF Write version	Uint16
150	2	Reserved	#00 bytes
152	L_IU	Implementation Use	bytes

152 + L_IU	4	VAT entry 0	Uint32
156 + L_IU	4	VAT entry 1	Uint32
…	…	…	…
Information Length - 4	4	VAT entry n	Uint32

The Length of Header field indicates the amount of data preceding the VAT entries. This value shall be 152 + L_IU.

The Length of Implementation use field shall specify the number of bytes in the Implementation Use field. If this field is non-zero, the value shall be at least 32 and be an integral multiple of 4.

The Logical Volume Identifier field shall identify the logical volume. This field shall be used by implementations instead of the corresponding field in the Logical Volume Descriptor. The value of this field should be the same as the field in the LVD until changed by the user.

The Previous VAT ICB location field shall be the logical block number of an earlier VAT ICB in the partition identified by the partition map entry. If this field is #FFFFFFFF, no such ICB is specified.

The Number of FIDs identifying Files identifies the number of files on the volume, including hard links. The number of files includes all FIDs in the heirarchy for which the directory bit is not set. The count does not include FIDs with the deleted bit set to one. The contents of this field shall be used by implementations instead of the corresponding field in the LVID.

The Number of non-parent FIDs identifying Directories identifies the number of directories on the volume, plus the root directory. The count does not include FIDs with the deleted bit set to one. The contents of this field shall be used by implementations instead of the corresponding field in the LVID.

The Min UDF Read version is defined in [LVID]. The contents of this field shall be used by implementations instead of the corresponding field in the LVID.

The Min UDF Write version is defined in [LVID]. The contents of this field shall be used by implementations instead of the corresponding field in the LVID.

The Max UDF Write version is defined in [LVID]. The contents of this field shall be used by implementations instead of the corresponding field in the LVID.

The Implementation Use field, if non-zero in length, shall begin with a regid identifying the usage of the remainder of the Implementation Use area.

VAT entry n shall identify the logical block number of the virtual block n. An entry of #FFFFFFFF indicates that the virtual sector is currently unused. The LBN specified is located in the partition identified by the partition map entry. The number of entries in the table can be determined from the VAT file size in the ICB:

Number of entries (N) = (Information Length - L_HD) / 4.

OSTA
Optical Storage
Technology Association

Document Change Notice
2-057

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Clarification on virtual space*
Date: September 29, 1997

Description:

Restrictions to help management of virtual space

Add at end of section 2.3.10:

Allocation Descriptors identifying virtual space shall have an extent length of the block size or less. Allocation descriptors identifying file data, directories, or stream data shall identify physical space. ICBs recorded in virtual space shall use long_ad allocation descriptors to identify physical space. The use of short_ad allocation descriptors would identify file data in virtual space if the ICB is in virtual space.
Descriptors recorded in virtual space shall have the virtual logical block number recorded in the Tag Location field.

OSTA
Optical Storage
Technology Association

Document Change Notice
2-058

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *File Times Extended Attribute*
Date: October 7, 1997

Description:

The restrictions in UDF 1.50 that limited the *File Times Extended Attribute* to the file creation time shall be removed; all four file times are now allowed. Also, the precedence of a file creation time in this Extended Attribute and the Creation Date and Time field of the main Extended File Entry is clarified.

Change:

Remove section 3.3.4.3.1 Uint32 FileTimeExistence; remove the highlighting of this field in section 3.3.4.3.

Modify section 3.3.4.3.2 as follows:

3.3.4.3.1 byte FileTimes

- If this field contains a file creation time it shall be interpreted as the creation time of the associated file. If the main *File Entry* is an *Extended File Entry*, the file creation time in this structure shall be ignored and the file creation time from the main *File Entry* shall be used.

- If the main File Entry is an Extended File Entry, this structure shall not be recorded with a file creation time.

If the main *File Entry* is not an *Extended File Entry* and the File Times Extended Attribute does not exist or does not contain the file creation time then an implementation shall use the *Modification Time* field of the *File Entry* to represent the file creation time.

OSTA
Optical Storage
Technology Association

Document Change Notice
2-059

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Macintosh Permissions*
Date: October 7, 1997

Description:

There is a problem browsing a UDF disc on Macintosh systems. If the applicable UDF *read* and *execute* permissions bits are not set, the contents of the directory cannot be displayed. On the Macintosh there is no way for a user to *unlock* the directory by setting the *execute* bit. Therefore on the Macintosh the implementation may ignore the status of the *execute* bit in regards to directory access.

Change:

Change the *Processing Permissions* table to indicate an "I" for ignore in regards to the Read permission of a directory under the Macintosh operating system.

Modify the paragraph following the table as follows:

"To be able to list the contents of a directory both the *Read* and *Execute* permission bits must be set for the directory. To be able to create, delete and rename a file or subdirectory both the *Write* and *Execute* permission bits must be set for the directory.
To get a better understanding of the *Execute* bit for a directory reference any UNIX book that covers file and directory permissions. The rules defined by the *Execute* bit for a directory shall be enforced by all implementations. The exception to this rule applies to Macintosh implementations. A Macintosh implementation may ignore the status of the *Read* bit in determining the accessibility of a directory."

OSTA
Optical Storage
Technology Association

Document Change Notice
2-060

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Case Sensitivity in File Identifier Lookup*
Date: October 7, 1997

Description:

Some operating systems are case insensitive in regards to file identifiers. A case insensitive operating system would consider the identifier "ABC.TXT", "Abc.Txt" and "abc.txt" to be identifiers for the same file. Because of this feature this type of operating system would not be able to access two separate files with identifiers that differ only by case. Certain of the operating systems that are case insensitive are also *case retentive* in regards to the filename that gets passed to the file system from the user. A *case retentive* operating system preserves the case in which the user originally used for the file identifier.

Operating systems which are *case insensitive* and *case retentive* include OS/2, Macintosh, Windows 9X and Windows NT. When performing directory searches UDF implementations for these operating systems should first perform a *case sensitive* search, and if that fails then perform a *case insensitive* search. This allows a user on the above mentioned operating systems access to separate files with identifiers that differ only by case.

Change:

In 4.2.2.1.2, 4.2.2.1.3, and 4.2.2.1.4 the following change will be made:

1. *FileIdentifier* Lookup: Upon request for a *"lookUp"* of a *FileIdentifier*, a case-sensitive comparison should be performed. If the case-sensitive comparison fails, a case-insensitive comparison should be performed.

OSTA
Optical Storage
Technology Association

Document Change Notice
2-061

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *OS/2 EA Stream*
Date: October 7, 1997

Description:

With the addition of stream support OS/2 Extended Attributes are better represented as a stream.

Change:

Change 3.3.4.5.3 as follows:

OS/2 supports an unlimited number of extended attributes which shall be stored as a named stream as defined in 3.3.8.2. To enhance performance the following *Implementation Use Extended Attribute* will be created.

Delete section 3.3.4.5.3.1.

Renumber 3.3.4.5.3.2, and change it as follows:

3.3.4.5.3.1 OS2EALength

This attribute specifies the OS/2 Extended Attribute Stream (3.3.8.2) information length. Since this value needs to be reported back to OS/2 under certain directory operations, for performance reasons it *should* be recorded in the *ExtendedAttributes* field of the *FileEntry*. This extended attribute shall be stored as an *Implementation Use Extended Attribute* whose *ImplementationIdentifier* shall be set to:

"***UDF OS/2 EALength**"

The *ImplementationUse* area for this extended attribute shall be structured as follows:

OS2EALength format

RBP	Length	Name	Contents
0	2	Header Checksum	Uint16
2	4	OS/2 Extended Attribute Length	Uint32

The value recorded in the *OS2ExtendedAttributeLength* field shall be equal to the *InformationLength* field of the file entry for the ***OS2EA*** stream.

Add the following the section on streams:

3.3.8.2 OS2EA Stream

All OS/2 definable extended attributes shall be stored as a named stream whose name shall be set to:

"***UDF OS/2 EA**"

The *OS2EA Stream* contains a table of OS/2 Full EAs (*FEA)* as shown below.

FEA format

RBP	Length	Name	Contents
0	1	Flags	Uint8
1	1	Length of Name (=L_N)	Uint8
2	2	Length of Value (=L_V)	Uint16
4	L_N	Name	bytes
4+L_N	L_V	Value	bytes

For a complete description of Full EAs (*FEA*) please reference the following IBM document:

"Installable File System for OS/2 Version 2.0"
OS/2 File Systems Department
PSPC Boca Raton, Florida
February 17, 1992

OSTA
Optical Storage
Technology Association

Document Change Notice
2-062

Document: OSTA Universal Disk Format Specification
Revision 1.50

Subject: *Non-Allocatable Space Stream*
Date: August 27, 1997

Description:

The Non-Allocatable Space List shall be recorded as a system stream.

Change:

Delete section 2.3.13.

Add the following to the section on streams:

3.3.7.2 Non-Allocatable Space Stream

ECMA 167 does not provide for a mechanism to describe defective areas on media or areas not usable due to allocation outside of the file system. The *Non-Allocatable Space Stream* provides a method to describe space not usable by the file system. The *Non-Allocatable Space Stream* shall be recorded only on media systems that do not do defect management (eg. CD-RW).

The *Non-Allocatable Space Stream* shall be generated at format time. All space indicated by the *Non-Allocatable Space Stream* shall also be marked as allocated in the free space map. The *Non-Allocatable Space Stream* shall be recorded as a named stream in the system stream directory of the *File Set Descriptor*. The stream name shall be:

"*UDF Non-Allocatable Space"

The stream shall be marked with the attributes *Metadata* (bit 4 of file characteristics set to ONE) and System (bit 10 of ICB flags field set to ONE). This stream shall have all Non-Allocatable sectors identified by its allocation extents. The allocation extents shall indicate that each extent is allocated but not recorded. This list shall include both defective sectors found at format time and space allocated for sparing at format time.

OSTA
Optical Storage
Technology Association

Document Change Notice
2-063

Document:	OSTA Universal Disk Format Specification Revision 1.50
Subject:	*Clarification on FIDs and the deleted bit*
Date:	September 29, 1997

Description:

Details on deleting files

Add section 2.3.4.2:

The deleted bit may be used to mark a file or directory as deleted instead of removing the FID from the directory, which requires rewriting the directory from that point to the end. If the space for the file or directory is deallocated, the implementation shall set the extent length of the ICB field to zero, as all fields in a FID must be valid even if the deleted bit is set. See [4/14.4.3], note 21 and [4/14.4.5].

No two FIDs in a directory shall have the same File Identifier (and File Version Number, which shall be 1), regardless of the state of the deleted bits of those FIDs. See [4/8.6]. Note: Implementations should re-use FIDs with the deleted bit set to one and extent lengths set to zero to avoid growing the size of the directory.

When deleting a File Identifier Descriptor an implementation may change the Compression ID to 0xFF and set the next four or eight bytes of the identifier to the byte offset of the FID within the directory as a Uint32 or Uint64 value. L_FI shall be set to 5 or 9. During scans of the directory, FIDs with a compression ID of 0xFF may be ignored.

OSTA
Optical Storage
Technology Association

Document Change Notice
2-064

Document:	OSTA Universal Disk Format Specification Revision 1.50
Subject:	*Clarification of DVD-Video Constraints*
Date:	March 20, 1998

Description:

This DCN clarifies the restrictions imposed on the UDF format when used for DVD-Video.

Change the second paragraph of section 6.9.1

from:

All DVD-Video discs shall be mastered to contain all required data as specified by ECMA 167 and UDF. This will ease playing of DVD-Video in computer systems. Examples of such data include the time, date, permission bits, and a free space map (indicating no free space). While DVD player implementations may ignore these fields, a UDF computer system implementation will not. Both entertainment-based and computer-based content can reside on the same disc.

to:

All DVD-Video discs shall be mastered to contain all required data as specified by ECMA 167 (2nd edition) and UDF 1.02. This will ease playing of DVD-Video in computer systems. Examples of such data include the time, date, permission bits, and a free space map (indicating no free space). While DVD player implementations may ignore these fields, a UDF computer system implementation will not. Both entertainment-based and computer-based content can reside on the same disc.

NOTE: DVD-Video discs mastered according to UDF 2.00 may not be compatabile with the DVD-Video players. DVD-Video players expect media in UDF 1.02 format.