Optical Storage Technology Association

UDF Document Change Notice
DCN-5049

Document: OSTA Universal Disk Format **DCN-5049**
Subject: *FID File Identifier length and Unicode uniqueness.*
Date: may 1, 2001

Description:

UDF 2.01 has redefined the interpretation of the special compression codes for deleted FIDs. What is still missing however is the fact that the File Identifier of a FID shall be unique, even after uncompress.
Further a value of 1 should not be allowed for the 'Length of File Identifier' field because this will uncompress to an empty string and causes confusion with a value 0 for this field which is only allowed in a parent FID. Further, a value of 1 is also not allowed in a Dstring.
The rule for 'Length of File Identifier' is put in the 'File Identifier' paragraph instead of introduction of a new paragraph for 'Length of File Identifier' and consequent renumbering of existing paragraphs and references.
This DCN is for the next UDF release after 2.01 and Errata for UDF 2.01.

Change:

In **2.3.4 File Identifier Descriptor**
change: char FileIdentifier[];
to: char **FileIdentifier[];**

add new paragraph:

2.3.4.6 char FileIdentifier[];

Contains a File Identifier stored in the OSTA Compressed Unicode format, see 2.1.1. The byte length of this field shall be greater than 1 with the sole exception of 0 for a parent FID. If the deleted bit is set in the File Characteristics field of this File Identifier Descriptor, then see 2.3.4.2 for additional rules. If the deleted bit is not set, then the Unicode representation of the File Identifier shall be unique in this directory. This requires not only byte-wise uniqueness as required by ECMA 4/8.6, but also uniqueness of the Unicode identifier resulting from uncompress of the OSTA Compressed Unicode format.

OSTA
Optical Storage
Technology Association

UDF Document Change Notice
DCN-5061

Document: OSTA Universal Disk Format DCN-5061
Subject: *Disallow overlapping partitions.*
Date: April 27, 2001

Description:

ECMA 3/8.7 offers the possibility of Partition Descriptors with overlapping partitions. As decided in the Feb 2001 TC meeting, UDF will not allow and never intended to allow overlapping partitions. Another error in the section at hand is the fact that WORM is not an existing Partition Access Type according to ECMA 3/10.5.7 and must be replaced by Write-Once. This DCN is for the next UDF release after 2.01 and Errata of previous releases.

Change:

In UDF 2.01, replace the Partition Descriptor section of 2. Basic Restrictions & Requirements at the bottom of page 8 and top of page 9.

Replace by:

A Partition Descriptor Access Type of Read-Only, Rewritable, Overwritable and Write-Once shall be supported. There shall be exactly one prevailing Partition Descriptor recorded per volume, with one exception. For Volume Sets that consist of single volume, the volume may contain 2 non-overlapping Partitions with 2 prevailing Partition Descriptors only if one has an access type of Read-Only and the other has an access type of Rewritable, Overwritable, or Write-Once. The Logical Volume for this volume would consist of the contents of both partitions.

OSTA
Optical Storage Technology Association

UDF Document Change Notice
DCN-5062

Document: OSTA Universal Disk Format DCN-5062
Subject: *Strategy 4096 obsolete for non-WORM media.*
Date: April 27, 2001, re-opened March 14, 2003.

Description:

Strategy type 4096 seems to be obsolete except for WORM implementations that want to upgrade to the newer UDF releases. Furthermore, it is hard to combine strategy type 4096 with a Metadata Partition (see DCN-5086) and it is better to define WORM in terms of partition Access Type. Therefore strategy 4096 will only be allowed for non-sequential write-once partitions for UDF 2.50 and higher revisions.
This DCN is for the next UDF release after 2.01.

Change:

Replace in 2.3.5.1:

for write: Shall be set to 4 or 4096.

by:

for write: Shall be set to 4 or 4096, see NOTE.

Replace the NOTE: in 2.3.5.1 by:

NOTE: Strategy type 4096, which is defined in the appendix, is intended for use on WORM media. Strategy type 4096 is allowed only for ICBs in a partition with Access Type write-once on non-sequential write once media.

OSTA
Optical Storage Technology Association

UDF Document Change Notice
DCN-5063

Document: OSTA Universal Disk Format DCN-5063
Subject: *UDF Unique ID Mapping Data*
Date: June 18, 2001, editorial update, March 7, 2003.

Description:

The sections 3.3.7.1 and 3.3.7.1.* are unclear and inconsistent on some points. It is not defined how an entry is marked as unused, e.g. in case an object is deleted. Further it is not clear where blank entries are allowed and under which Flags bit 0 conditions.

Editorial remark: Because changing of the UniqueID during a rename or move operation was disallowed in the approved DCN-5078, DCN-5063 is updated accordingly.

Change:

Replace sections 3.3.7.1 and 3.3.7.1. by:*

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 UDF Unique ID, or to the ICB hierarchy for the directory that contains the file/directory associated with the UDF Unique ID. Note that for UDF release 2.00 and higher the UDF Unique ID value used for this purpose is taken from the File Identifier Descriptor rather than from the File Entry.

Unique ID Mapping Data is stored as a named stream of the *System Stream Directory* (associated with the File Set Descriptor). 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 for the stream shall be set to 1 to indicate that the existence of this stream should not be made known to clients of a platform's file system interface.
Rules for the presence and consistency of the Unique ID Mapping Data Stream:

- Shall be created for read-only media
- Shall be created by implementations with 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 Allocation Table is recorded.

3.3.7.1.1 UDF Unique ID Mapping Data

The contents of the Unique ID Mapping Stream are described by the tables "UDF Unique ID Mapping Data" and "UDF Unique ID Mapping Entry". The mapping data contains some header fields before an array of mapping entries. The fields of these structures are described below their corresponding table.

---
unchanged table: **UDF Unique ID Mapping Data**

Implementation Identifier is described in section 2.1.5.

Flags are defined as follows:

Bit 0	Index Bit
Bits 1 – 31	Reserved, shall be set to ZERO

Index Bit set to ONE is called Index Mode. In Index Mode, the UDF Unique ID, once decremented by 16 (the value NextUniqueID is initialized to), can be used as an index into the array Mapping Entries.

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.

3.3.7.1.2 UDF Unique ID Mapping Entry

---
unchanged table: **UDF Unique ID Mapping Entry**

UDF Unique ID is the value found in the FID identifying the object.

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 from the long_ad ICB field of the FID identifying the object.

Parent Partition Reference Number is the partition reference number of the ICB identifying the directory that contains the FID identifying the object.

Object Partition Reference Number is the partition reference number from the long_ad ICB field of the FID identifying the object.

In Index Mode, the first entry has a UDF Unique ID of 16 and subsequent entries are required to have a UDF Unique ID value of one more than the preceding entry.

If not in Index Mode, invalid entries may be removed in order to shrink the array.

Invalid entries are represented by having a value of zero in all fields, except the UDF Unique ID field. Invalid entries are the result of objects that were deleted from the medium or entries at the end of the Mapping Entries array that are not yet in use.

There shall only be valid entries for non-stream, non-parent FIDs.

NOTE: The UDF Unique ID value of a mapping entry for an object needs not be equal to the Unique ID value found in the File Entry of the object.

The correctness of a mapping entry can be verified performing the following steps:

1. Read the File Entry of the parent directory of the object using the Parent Logical Block Number and the Parent Partition Reference Number of the mapping entry.
2. Find in the parent directory a FID with a UDF Unique ID value equal to the UDF Unique ID of the mapping entry.
3. The long_ad ICB field of this FID shall contain logical block number and partition reference number values equal to the Object Logical Block Number and Object Partition Reference Number values of the mapping entry respectively.

OSTA
Optical Storage
Technology Association

UDF Document Change Notice
DCN-5064

Document: OSTA Universal Disk Format DCN-5064
Subject: *Extended Attribute block alignment.*
Date: April 27, 2001

Description:

UDF 3.3.4: It is impossible to block align the start of an ECMA 167 Extended Attribute if it is the only ECMA 167 EA in an EA space because it starts directly after the EA Header Descriptor. At the March 2001 TC meeting it was decided not to require start block alignment for ECMA 167 EAs. This can be restricted to the first one only.
This DCN is for the next UDF release after 2.01 and errata of previous releases.

Change:

In **3.3.4 Extended Attributes**

change:

1. *All* EAs with an attribute length greater than or equal to a logical block shall be block aligned by starting and ending on a logical block boundary.

into:

1. *All* EAs with an attribute length greater than or equal to a logical block shall be block aligned by starting and ending on a logical block boundary. The one and only exception to this rule is the start of the first ECMA 167 EA.

OSTA
Optical Storage
Technology Association

UDF Document Change Notice
DCN-5065

Document: OSTA Universal Disk Format DCN-5065
Subject: *UDF Defined Named Streams section.*
Date: May 3, 2001

Description:

Section "3.3.5.2 System Named Streams (Metadata)" causes confusion because of inconsistent terminology and different terminology compared to subsequent sections 3.3.7 and 3.3.8. Further the sentence defining the file type for streams is valid for all named streams and not special for UDF named streams or system streams. The rule is already covered in 3.3.5.1 for all streams and can be removed from 3.3.5.2.
This DCN is for the next UDF release after 2.01 and Errata for UDF 2.00 and 2.01.

Change:

replace the first part of section 3.3.5.2
till "The four characters *UDF…"
by:

3.3.5.2 UDF Defined 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 (System Stream Directory) and apply to the entire file set. These are called UDF Defined System Streams and are defined in section 3.3.7. Others pertain to individual files or directories and are identified by the Stream Directory of that particular file or directory. These are called UDF Defined Non-System Streams and are defined in 3.3.8.

All UDF Defined Named Streams shall have the Metadata bit set in the File Identifier Descriptor in the Stream Directory, unless otherwise specified in this document. All streams not generated by the file system implementation shall have this bit set to zero.

The four characters *UDF …

Optical Storage Technology Association

UDF Document Change Notice
DCN-5066

Document: OSTA Universal Disk Format DCN-5066
Subject: *FileIdentifier translation code repair, 6.7.2.*
Date: June 19, 2001

Description:

The FileIdentifier translation algorithm code for non-DOS Operating Systems still contains an error.

Change:

in **6.7.2 OS/2, Macintosh,Windows 95, Windows NT and UNIX Algorithm**

replace at the top of page 125:

```
    || !isprint(udfName[extIndex + index + 2])))
```

by:

```
    || ! UnicodeIsPrint(udfName[extIndex + index + 2])))
```

OSTA Optical Storage Technology Association

UDF Document Change Notice
DCN-5068

Document:	OSTA Universal Disk Format DCN-5068
Subject:	*Editorial changes for UDF revision after UDF 2.01*
Date:	May 24, 2001. Last update, March 20, 2003

Subject:	*Add section title for FreeAppEASpace extended attribute*
Date:	May 24, 2001
Source:	Victor Peng

Description:

In section 3.3.4.6.1 of the UDF Spec 2.01, a title for its FreeAppEASpace sub-section is missed.

Change:

In section 3.3.4.6.1(Application Use Extended Attributes) (All Operating Systems),

add a line:

3.3.4.6.1.1 FreeAppEASpace

under the section title "All Operating Systems".

Subject:	*Synchronise ECMA and UDF structure field names.*
Date:	7/26/01
Source:	Gerrit Scholl

Description

It is very confusing to have field names
in UDF that differ from the ECMA names.

Change

in: 2.3.5 ICB Tag
and: 3.3.2 ICB Tag
and: 4.2.1 ICB Tag

replace: Uint16 NumberofEntries;
by: Uint16 MaximumNumberofEntries;

Subject:	*Fix media type*
Date:	8/13/01
Source:	Gerrit Scholl

In the last bullet of: 6.10.1.1 Requirements

replace: CD media
by: CD-R media

Subject:	*Descriptor lengths table*
Date:	February 11, 2003
Source:	Gerrit Scholl

Add Sparing Table to table in 5.1, length in bytes “no max”.

Subject:	*OSTA UDF E-mail reflector commands*
Date:	February 11, 2003
Source:	Gerrit Scholl

For the “OSTA UDF E-Mail Reflector”, the “subscribe udf” and “unsubscribe udf” commands must be sent as a line in the mail text, not in the subject.

Subject:	*Rephrase in 3.3.7.2*
Date:	February 17, 2003
Source:	Tom Jolly

3.3.7.2 para 3 - rephrase?
I'd suggest changing "allocation extents" to "allocation descriptors"
and sectors to blocks (since we're in the lvol)

in 3.3.7.2 replace:

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.

by:

The stream's allocation descriptors shall identify all non-allocatable packets. The allocation descriptors shall have allocation type 1 (allocated but not recorded).

Subject:	*Misc*
Date:	February 17, 2003
Source:	Mat Timmermans

In 2.1.5.2 for Device Specification Extended Attribute field name:

replace: Implementation ID
by: Implementation Use

Subject:	*SystemStreamDirectory field*
Date:	March 6, 2003
Source:	Gerrit Scholl

in 2.3.2 replace: struct long_ad StreamDirectoryICB;
by: struct long_ad SystemStreamDirectoryICB;

Subject:	*Editorial DCNs on UniqueID.*
Date:	March 7, 2003
Source:	Gerrit Scholl

The DCNs 5063, 5072 and 5078 are all involved with UniqueIDs. 5078 was as last approved in the Feb 2003 meeting.
The updated text by DCN-5072 was moved to a different section because of changes requested by DCN-5078.
DCN-5078 disallows change of UniqueID in a rename or move operation. DCN-5063 had to be adapted accordingly.

Editorial notes are added to all DCNs involved with these changes.

Subject:	*FreeSpace Table and Size Table values*
Date:	March 13, 2003
Source:	Yoshiho Gotoh

in 2.2.6.2. replace: the FreeSpaceTable shall be set
by: the FreeSpaceTable value shall be set

in 2.2.6.3. replace: the SizeTable shall be set
by: the SizeTable value shall be set

Subject:	*Arrays in section header*
Date:	March 13, 2003
Source:	Gerrit Scholl

Throughout the document, all section headers explaining array fields should show in the field explanation section header that an array is involved:

e.g. in 4.2.2.1. replace: char FileIdentifier
by: char FileIdentifier[]

Subject:	*DVD Consortium corrections*
Date:	March 14, 2003
Source:	Yoshiho Gotoh

in 3.3.4.5.1.2 replace: DVD Consortium (see 6.9.3).

by: DVD Format/Logo Licensing Corporation, see 6.9.3.

in 6.9.1, replace:

NOTE: … developed by the DVD Consortium, that describes …

by:

NOTE: … published by the DVD Format/Logo Licensing Corporation, see 6.9.3. This document describes …

Subject:	*EntityID definitions and references to 2.1.5.*
Date:	March 14, 2003
Source:	Gerrit Scholl, Yoshiho Gotoh

Description:

In several cases where section 2.1.5.3 is referenced, a reference to the table in 2.1.5.2 would be more appropriate, except where the hard or soft protect bits are referenced. However the whole section 2.1.5 is important. So the EntityID references to 2.1.5.3 are changed to 2.1.5.
Further it appeared that there was confusion about how the Suffix Type for an EntityID is determined. It is not the name of the EntityID field, but the table of 2.1.5.2 that determines the Suffix Type (e.g. see the Implementation Identifier field of the IUVD that has an UDF Identifier Suffix). Also a note is added that UDF uses EntityID instead of the ECMA regid.

Changes:

rephrase 2.1.5 as follows:

2.1.5 Entity Identifier

```
struct EntityID {          /* ECMA 167 1/7.4 */
        Uint8              Flags;
        char               Identifier[23];
        char               IdentifierSuffix[8];
}
```

NOTE: UDF uses *EntityID* for the structure that is called *regid* in ECMA-167.

UDF classifies *Entity Identifiers* into 4 separate types. Each type has its own *Suffix Type* for the *IdentifierSuffix* field. The 4 types are:

- *Domain Entity Identifiers* with a *Domain Identifier Suffix*
- *UDF Entity Identifiers* with a *UDF Identifier Suffix*
- *Implementation Entity Identifiers* with an *Implementation Identifier Suffix*
- *Application Entity Identifiers* with an *Application Identifier Suffix*

The following sections describe the format and use of *Entity Identifiers* based upon the different types mentioned above. For all UDF descriptor fields containing an EntityID structure, the value of the *Identifier* field and the *Suffix Type* for the *IdentifierSuffix* field are defined in the Entity Identifiers table of 2.1.5.2. The interpretation of the *IdentifierSuffix* field for each *Suffix Type* is defined in 2.1.5.3.

change all EntytyID references to 2.1.5.2 and 2.1.5.3 into a reference to 2.1.5 and rephrase the ImplementationUse paragraph of 3.3.4.4 to:

As the first structure in the *ImplementationUse* field, an EntityID shall be recorded by all implementations. This EntityID uniquely identifies the current implementation by a developer ID, see 2.1.5.

Subject:	*Developer Registration Form update.*
Date:	March 14, 2003
Source:	Gerrit Scholl

change in 6.13 (now 6.16):

replace in first[t] bullet: … and their associated *Implementation Identifiers*.
by: … and their associated *Developer IDs*.

on form add option for OS support: Windows XP

on form add options for media support: CD-MRW, DVD+R, DVD+MRW

use Developer ID terminology in Developer Registration Form and refer to 2.1.5, replace:

Please indicate what value you plan to use in the *Implementation Identifier* field of the *Entity Identifier* descriptor to identify your implementation:

_____

NOTE: *The identifier should be something that uniquely identifies your company as well as your product.*

by:

Please indicate what value you plan to use as the EntityID *"*Developer ID"* **to identify your implementation, see 2.1.5:**

_____

NOTE: *The Developer ID should be something that uniquely identifies your company as well as your product.*

replace:

FAX Completed form to OSTA at 1-805-962-1541, or mail to:
OSTA, 311 E. Carrillo Street, Santa Barbara, CA 93101

by:

E-mail or fax completed form to OSTA. For address, see POINTS OF CONTACT on the first page of this document.

Subject: *Revision History Update*
Date: March 14, 2003
Source: Gerrit Scholl

Add to the table of 6.15:

Update of this table for UDF 2.01 still to be done	t.b.d.	t.b.d.	t.b.d.	t.b.d.
FID File Identifier length and Unicode uniqueness	5049	2.50	1.02	2.01
Disallow overlapping partitions	5061	2.50	1.02	1.02
Strategy 4096 only for WORM media	5062	2.50	1.02	1.02
UDF Unique ID Mapping Data	5063	2.50	2.50	2.50
Extended Attribute block alignment	5064	2.50	1.02	1.02
UDF Defined Named Streams section	5065	2.50	2.00	2.00
File Identifier translation code repair	5066	2.50	1.02	1.02
Correction of is_fileset_soft_protected rule	5069	2.50	2.00	2.00
Disallow hard linked directories	5070	2.50	1.02	2.50
Requirements for DVD-RAM/RW/R interchangeability	5071	2.50	2.00	2.00
Unique ID for System Stream Directory	5072	2.50	2.50	2.50
Shared description for some LVID and VAT fields	5074	2.50	2.01	2.01
Recommendations for Mount Rainier formatted media	5075	2.50	1.02	1.02
Recommendations for DVD+R and DVD+RW	5076	2.50	1.50	1.50
Section 3.3.6 put out of order	5077	2.50	2.00	2.00
UDF UniqueID clarifications	5078	2.50	2.00	2.00
Clarify partition Access Type 3 and 4	5079	2.50	2.01	2.01
Icbtag Parent ICB Location issue	5081	2.50	1.02	2.50
Clarification of Volume Recognition Sequence	5082	2.50	1.02	2.01
Metadata Partition Map	5086	2.50	2.50	2.50
Partition Alignment & ECC Block Size Definition	5089	2.50	1.02	2.50
Non-allocatable space stream usage clarifications	5090	2.50	1.50	1.50

Subject: *Misc.*
Date: March 17, 2003
Source: Gerrit Scholl, Tom Jolly

in 6.3, 2nd table (OS Identifiers)

replace: Windows NT – generic (includes Windows 2000)

by: Windows NT – generic (includes Windows 2000,XP,Server 2003, and later releases based on the same code base)

replace: Windows 9x – generic (includes Windows 98)

by: Windows 9x – generic (includes Windows 98/ME)

insert:

3	1	Macintosh OS X and later releases.

Replace: Macintosh OS

By: Macintosh OS 9 and older.

in 1.2 'Backwards Write Compatibility' and the NOTE of 6.9.1
replace: "2.0x" *by:* "2.xx"

Optical Storage Technology Association

UDF Document Change Notice
DCN-5068

Subject:	*Replace OSTA Legal Text*
Date:	March 20, 2003
Source:	Gerrit Scholl

replace OSTA Legal Text as received from OSTA.

in 'Important Notices' replace all text between the horizontal line through:

"Universal Disk Format® and UDF® are registered marks of the Optical Storage Technology Association."

by:

(a) THIS DOCUMENT IS AN AUTHORIZED AND APPROVED PUBLICATION OF OSTA. THE SPECIFICATIONS CONTAINED HEREIN ARE THE EXCLUSIVE PROPERTY OF OSTA BUT MAY BE REFERRED TO AND UTILIZED BY THE GENERAL PUBLIC FOR ANY LEGITIMATE PURPOSE, PARTICULARLY IN THE DESIGN AND DEVELOPMENT OF WRITABLE OPTICAL SYSTEMS AND SUBSYSTEMS. THIS DOCUMENT MAY BE COPIED IN WHOLE OR IN PART PROVIDED THAT NO REVISIONS, ALTERATIONS, OR CHANGES OF ANY KIND ARE MADE TO THE MATERIALS CONTAINED HEREIN.

(b) COMPLIANCE WITH THIS DOCUMENT MAY REQUIRE USE OF ONE OR MORE FEATURES COVERED BY THE PATENT RIGHTS OF AN OSTA MEMBER, ASSOCIATE OR THIRD PARTY. NO POSITION IS TAKEN BY OSTA WITH RESPECT TO THE VALIDITY OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT, WHETHER OWNED BY A MEMBER OR ASSOCIATE OF OSTA OR OTHERWISE. OSTA HEREBY EXPRESSLY DISCLAIMS ANY LIABILITY FOR INFRINGEMENT OF INTELLECTUAL PROPERTY RIGHTS OF OTHERS BY VIRTUE OF THIS OSTA DOCUMENT, NOR DOES OSTA UNDERTAKE A DUTY TO ADVISE USERS OR POTENTIAL USERS OF OSTA DOCUMENTS OF SUCH NOTICES OR ALLEGATIONS. OSTA HEREBY EXPRESSLY ADVISES ALL USERS OR POTENTIAL USERS OF THIS DOCUMENT TO INVESTIGATE AND ANALYZE ANY POTENTIAL INFRINGEMENT SITUATION, SEEK THE ADVICE OF INTELLECTUAL PROPERTY COUNSEL AND, IF INDICATED, OBTAIN A LICENSE UNDER ANY APPLICABLE INTELLECTUAL PROPERTY RIGHT OR TAKE THE NECESSARY STEPS TO AVOID INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT. OSTA EXPRESSLY DISCLAIMS ANY INTENT TO PROMOTE INFRINGEMENT OF ANY INTELLECTUAL PROPERTY RIGHT BY VIRTUE OF THE EVOLUTION, ADOPTION, OR PUBLICATION OF THIS OSTA DOCUMENT.

(c) ONE OR MORE PATENT HOLDERS HAVE FILED STATEMENTS OF WILLINGNESS TO GRANT A LICENSE, ON REASONABLE AND NONDISCRIMINATORY TERMS, ON A RECIPROCAL BASIS, UNDER PATENT CLAIMS ESSENTIAL TO IMPLEMENT THIS SPECIFICATION. FURTHER INFORMATION MAY BE OBTAINED FROM OSTA.

(d) OSTA MAKES NO REPRESENTATION OR WARRANTY REGARDING ANY SPECIFICATION, AND ANY COMPANY USING A SPECIFICATION SHALL DO SO AT ITS SOLE RISK, INCLUDING SPECIFICALLY THE RISKS THAT A PRODUCT DEVELOPED WILL NOT BE COMPATIBLE WITH ANY OTHER PRODUCT OR THAT ANY PARTICULAR PERFORMANCE WILL NOT BE ACHIEVED. OSTA SHALL NOT BE LIABLE FOR ANY EXEMPLARY, INCIDENTAL, PROXIMATE OR CONSEQUENTIAL DAMAGES OR EXPENSES ARISING FROM THE USE OR IMPLEMENTATION OF THIS DOCUMENT. THIS DOCUMENT DEFINES ONLY ONE APPROACH TO COMPATIBILITY, AND OTHER APPROACHES MAY BE AVAILABLE IN THE INDUSTRY.

Universal Disk Format® and UDF® are registered marks of the Optical Storage Technology Association.

Subject:	*OSTA contact information.*
Date:	March 20, 2003
Source:	Hans Mons

In order to avoid address information that is not up-to-date, we refer to the contact information on the OSTA web site as much as possible. This information is in the "POINTS OF CONTACT" on the first page of the document. Section 5.5 "Technical Contacts" is removed and from the Developer Registration form, etc. a reference to "POINTS OF CONTACT" is made.

*replace '**POINTS OF CONTACT'** information by:*

Optical Storage Technology Association
http://www.osta.org

Contact information
http://www.osta.org/osta/contact.htm

Technical Editor
editor.udf@osta.org

Technical questions
info@osta.org

OSTA UDF E-Mail Reflector

To subscribe: address request@list.osta.org with "subscribe udf" as a line in the mail text.

To unsubscribe: address request@list.osta.org with "unsubscribe udf" as a line in the mail text.

Send messages to UDF reflector: udf@list.osta.org

Subject:	*2.3.6.4 InformationLength.*
Date:	March 20, 2003
Source:	Tom Jolly

This paragraph relating to reconstruction of information length is practically useless, since there is no guarantee of the order in which modifications will hit the media, and thus no reliable way to determine if the information length or the Ads represent the 'real' state of the file.

Remove this paragraph.

Subject:	*1.3.4 Acronyms – SBD missing*
Date:	March 20, 2003
Source:	Tom Jolly

Add definition for SBD to table 1.3.4 Acronyms.

SBD	Space Bitmap Descriptor

Subject: *Allocation Extent Descriptor CRC exception missing from section 2.*
Date: March 20, 2003
Source: Tom Jolly

The CRC is optional for the allocation extent descriptor. This is missing from *2. Basic Restrictions & Requirements.*

Replace:

Descriptor CRCs — CRCs shall be supported and calculated for all Descriptors, except for the Space Bitmap Descriptor.

by:

Descriptor CRCs	CRCs shall be supported and calculated for all Descriptors except for the Space Bitmap Descriptor. There is a CRC length special case for the Allocation Extent Descriptor.

Subject: *Final to-be-done issues.*
Date: March 20, 2003
Source: Gerrit Scholl

- *Remove "draft" on several places in text and footer*
- *Check date on front and next page(revision history) and in footer*
- *Check all web links and e-mail address links to: osta.org*
- Check all strange and unintended page breaks
- Regenerate final table of contents.
- Regenerate final word list.

Optical Storage Technology Association

UDF Document Change Notice
DCN-5069

Document: OSTA Universal Disk Format DCN-5069
Subject: *Error in 2.1.5.3, is_fileset_soft_protected rule.*
Date: October 17, 2001

Description:

This DCN is for the next UDF release after UDF 2.01 and errata for UDF 2.00 and 2.01. In 2.1.5.3, there is an error in the ‘is_fileset_soft_protected‘ rule.

Change:

in section 2.1.5.3 replace:

is_fileset_soft_protected = (LVD.SoftWriteProtect || FSD.SoftWriteProtect) &&
(!is_vol_hard_protected)

by:

is_fileset_soft_protected = (LVD.SoftWriteProtect || FSD.SoftWriteProtect) &&
!is_fileset_hard_protected

Optical Storage Technology Association

UDF Document Change Notice
DCN-5070

Document: OSTA Universal Disk Format DCN-5070
Subject: *Disallow hard linked directories*
Date: Mar 19, 2002

Description:

There are no direct rules for hard linked directories in UDF. ECMA 4/8.6 seems to allow hard linked directories if they have the same parent. I cannot imagine that this was the intention for UDF, so I propose not to allow hard linked directories at all.
Mind that identification of directories by a parent FID is not affected.

Change:

In **2.3.6 File Entry**
replace:

Uint16 FileLinkCount;

by:

Uint16 FileLinkCount;

Add new section:

2.3.6.8 FileLinkCount
Hard links to a directory are not allowed. A directory File Entry shall be identified by:
- for non-root directories: exactly one FID defining the directory name
- zero or more parent FIDs if appropriate.

For stream and stream directory hard link restrictions, see 3.3.5.1.

Optical Storage Technology Association

UDF Document Change Notice
DCN-5071

Document:	**OSTA Universal Disk Format DCN-5071** Revision 2.00 and 2.01
Subject:	Requirements for DVD-RAM/RW/R interchangeability
Date:	**November 13, 2000**

Description:

Requirements for DVD-RAM, DVD-RW, and DVD-R discs to be used with consumer appliances (e.g. dedicated DVD content recorder/player) are specified as a new appendix for UDF 2.00 and 2.01 to improve data interchangeability among these appliances and computer systems.

This text will be added to the next UDF revision after UDF 2.01 and is errata for UDF 2.00 and 2.01.

Change:

Add new appendix 6.12 as:

6.12 Requirements for DVD interchangeability

This appendix defines the requirements and restrictions on volume and file structures for writable DVD media, including but not limited to DVD-RAM discs (6.12.1), DVD-RW discs (6.12.2) and DVD-R discs (6.12.3), to support the interchange of information between users of both computer systems and consumer appliances. These requirements do not apply to the discs that are used in a computer system environment only and have no interchangeability with consumer appliances. The common requirements for these DVD discs are summarized as follows:

1. The volume and file structure shall comply with UDF 2.00.
2. The Minimum UDF Read Revision and Minimum UDF Write Revision shall be 2.00.
3. The length of logical sector and logical block shall be 2048 bytes.
4. A Main Volume Descriptor Sequence and a Reserve Volume Descriptor Sequence shall be recorded.

6.12.1 Requirements for DVD-RAM

The requirements for DVD-RAM discs are based on UDF 2.00. The volume and file structure is simplified as for overwritable discs using non-sequential recording.

For Volume Structure:

1. A partition on a DVD-RAM disc shall be an overwritable partition specified as access type 4.
2. Virtual Partition Map and Virtual Allocation Table shall not be recorded.
3. Sparable Partition Map and Sparing Table shall not be recorded.

For File Structure:

4. Unallocated Space Table or Unallocated Space Bitmap shall be used to indicate a space set. Freed Space Table and Freed Space Bitmap shall not be recorded.
5. Non-Allocatable Space Stream shall not be recorded.

6.12.2 Requirements for DVD-RW

The requirements for DVD-RW discs under Restricted Overwrite mode are based on UDF 2.00. The volume and file structure is simplified as for rewritable discs using non-sequential recording.

For Volume Structure:

1. A disc shall consist of a single volume with a single sparable partition per side.
2. A Sparable Partition Map and Sparing Table shall be recorded.
3. Length of a packet shall be 16 sectors (32 KB) and the first sector number of a packet shall be an integral multiple of 16.
4. Virtual Partition Map and Virtual Allocation Table shall not be recorded.

For File Structure:

5. Unallocated Space Bitmap shall be used to indicate a space set. Unallocated Space Table, Freed Space Table and Freed Space Bitmap shall not be recorded.
6. Non-Allocatable Space Stream shall be recorded.
7. ICB Strategy type 4 shall be used.
8. Short Allocation Descriptors or the embedded data shall be recorded in the Allocation Descriptors field of the File Entry or Extended File Entry. Long Allocation Descriptors shall not be recorded in this field.

6.12.3 Requirements for DVD-R

The requirements for DVD-R discs under Disc at once recording mode and under Incremental recording mode are based on UDF 2.00. The volume and file structure is simplified as for write once discs using sequential recording.

For Volume Structure:

1. Length of a packet shall be an integral multiple of 16 sectors (32 KB) and the first sector number of a packet shall be an integral multiple of 16.
2. Sparable Partition Map and Sparing Table shall not be recorded.
3. Under Incremental recording mode, only one Open Integrity Descriptor shall be recorded in the Logical Volume Integrity Sequence.
4. Under Incremental recording mode, Virtual Partition Map shall be recorded.

For File Structure:

5. Unallocated Space Table, Unallocated Space Bitmap, Freed Space Table and Freed Space Bitmap shall not be recorded.
6. Only one File Set Descriptor shall be recorded.

7.Non-Allocatable Space Stream shall not be recorded.
8.Under Incremental recording mode, Virtual Allocation Table and VAT ICB shall be recorded.
9.Under Incremental recording mode, ICB Strategy type 4 shall be used.
10.Under Incremental recording mode, the VAT entries in VAT shall be assigned as follows:
- The virtual address 0 shall be used for File Set Descriptor.
- The virtual address 1 shall be used for the ICB of the root directory.
- The virtual addresses in the range of 2 to 255 shall be assigned for the File Entry of DVD_RTAV directory and File Entries of files under the DVD_RTAV directory.

6.12.4 Requirements for Real-Time file recording on DVD discs

DVD Video Recording specification defines the DVD specific sub-directory "DVD_RTAV" and all DVD specific files under the DVD_RTAV directory. DVD specific files consist of Real-Time files with the file type 249 and the related information files.

For Volume Structure:

1.For DVD-RAM/RW discs, a disc shall consist of a single volume with a single partition per side. For DVD-R discs, a disc shall consist of a single volume with a write once partition and a virtual partition per side.
2.For DVD-RW discs, First Sparing Table and Second Sparing Table shall be recorded.

For File Structure:

3.For DVD-RAM/RW discs, only Unallocated Space Bitmap shall be used.
4.For DVD-RW discs, the extent of Unallocated Space Bitmap should have the length of Space Bitmap Descriptor for the available Data Recordable area.
5. Consumer Content Recorders record all their data in a special subdirectory, DVD_RTAV, located in the root directory. The DVD_RTAV directory and its contents have special file system restrictions which are defined in DVD Specifications published from DVD Format/Logo Licensing Corporation. An implementation or application should not create or modify files in this directory unless it meets the restrictions defined by DVD Specifications specified above.

OSTA Optical Storage Technology Association

UDF Document Change Notice
DCN-5071

Subject: Change for DVD documents contact information
Date: November 1, 2000

Description:

Contact information to obtain DVD documents is changed.

Change:

In 6.9.3, replace the contact information with

DVD Format/Logo Licensing Corporation
Shiba Shimizu Bldg. 5F,
2-3-11 Shibadaimon, Minato-ku
Tokyo 105-0012,
Japan

TEL: +81-3-5777-2883
FAX: +81-3-5777-2884

Optical Storage Technology Association

UDF Document Change Notice
DCN-5072

Document:	OSTA Universal Disk Format Specification Revision 2.01
Subject:	Unique ID usage in the System Named Stream directory.
Date:	June 22, 2000. Editorial remark March 7, 2003

Description:

The last sentence of 3.3.5.1 states:

The Unique ID field of Named Streams and Stream Directories shall be the same as the Unique ID of the main data stream.

It is impossible to follow this requirement when dealing with the System Stream Directory since they do not have a main stream but are associated with the File Set Descriptor, which does not have a Unique ID.

Change:

In section 3.3.5.1, modify the restriction reading:
The Unique ID field of Named Streams and Stream Directories shall be the same as the Unique ID of the main data stream.

To:

The Unique ID field of Named Streams and Stream Directories shall be the same as the Unique ID of the main data stream except for those residing in the System Stream directory. For the System Stream directory, all Unique ID field contents shall be set to zero when writing and ignored when reading.

Editorial note: Due to changes in DCN-5078, the text above is moved to section 3.2.1.1 and adapted to the context of 3.2.1.1. The text is adapted to:

FIDs and File Entries of the System Stream Directory and of streams associated with the System Stream Directory shall use a UniqueID value of zero.

OSTA
Optical Storage
Technology Association

UDF Document Change Notice
DCN-5074

Document: OSTA Universal Disk Format DCN-5074
Subject: *Shared description for some LVID and VAT fields*
Date: March 20, 2002

Description:

For a File System using a VAT, several LVID ImplementationUse fields of 2.2.6.4 are overruled by identical fields in the VAT, see 2.2.10. However, naming and description for these fields was different in UDF 2.00. When going to UDF 2.01, most of the field naming was made equal for LVID and VAT, but some description details were lost, especially for "Number of Files" and "Number of Directories".
This proposal falls back to one description for the LVID fields in 2.2.6.4 with all the details. The corresponding VAT fields in 2.2.10 use identical naming and point back to 2.2.6.4 for field description.
The most detailed description for compare is in UDF 2.00 2.2.10.
This DCN is for the next UDF release after 2.01 and as erratum for 2.01 and 2.00.

Change:

In 2.2.6.4 byte ImplementationUse

after the table add:

NOTE: For a Sequential File System using a VAT, all field values above will be overruled by the corresponding VAT fields, except for the ImplementationID and Implementation Use fields, see 2.2.10.

replace the "Number of Files" paragraph by:

Number of Files - The current number of files in the Logical Volume, including hard links. The count includes all FIDs in the directory hierarchy for which the Directory bit, Parent bit and Deleted bit are all ZERO. FIDs identifying a stream are not included in the count. This information is needed by the Macintosh OS. All implementations shall maintain this information.

replace the "Number of Directories" paragraph by:

Number of Directories - The current number of directories in the Logical Volume, plus the root directory. The count includes the root directory and all FIDs in the directory hierarchy for which the Directory bit is ONE and the Parent bit and Deleted bit are both ZERO. FIDs identifying a stream directory are not included in the count. This

information is needed by the Macintosh OS. All implementations shall maintain this information.

In 2.2.10 Virtual Allocation Table

in the VAT structure table,
in 3 places replace: Version
by: Revision

after the table replace description for "Number of files -"

till "Max UDF Write Version -" included

by:

Number of Files – Defined in 2.2.6.4.
The contents of this field shall be used instead of the corresponding LVID field.

Number of Directories - Defined in 2.2.6.4.
The contents of this field shall be used instead of the corresponding LVID field.

Minimum UDF Read Revision - Defined in 2.2.6.4.
The contents of this field shall be used instead of the corresponding LVID field.

Minimum UDF Write Revision - Defined in 2.2.6.4.
The contents of this field shall be used instead of the corresponding LVID field.

Maximum UDF Write Revision - Defined in 2.2.6.4.
The contents of this field shall be used instead of the corresponding LVID field.

OSTA
Optical Storage Technology Association

UDF Document Change Notice DCN-5075

Document: OSTA Universal Disk Format DCN-5075
Subject: *Recommendations for Mount Rainier formatted media (UDF 1.02 and higher)*
Date: Sept 10, 2002

Description:

This DCN defines recommendations for Mount-Rainier formatted CD-RW and DVD+RW media, referenced as CD-MRW and DVD+MRW media. This DCN is for new versions of UDF and is an erratum for UDF 1.02 until UDF 2.01.

Change:

Add new appendix 6.x (will be 6.14):

6.x Recommendations for Mount Rainier formatted media

The following guidelines are established to ensure interchange of Mount Rainier (MRW) formatted media.

6.x.1 Properties of CD-MRW and DVD+MRW media and drives

The following is a list of key properties of MRW media and drives:

- A Physical Sector Size of 2048 Bytes
- The drive performs Read/Modify/Write cycles when needed. Data transfer between the host and the MRW drive is in multiples of 2048 bytes.
- Random access read and write is possible
- Drive level defect management
- The drive performs background physical formatting
- The Media Type is Overwritable (partition access type 4)
- A Non-Allocatable Space List, Non-Allocatable Space Stream and Sparing table shall not be used on MRW formatted media

6.x.2 Background Physical Formatting

At the initialization of the file system, after the Background Physical Formatting has been started, the host must record the first AVDP at sector 256. The second AVDP must be recorded after the Background physical Formatting has been finished. Before the second AVDP has been recorded, the file system is in an intermediate state and is not strictly in compliance with ECMA 167. The disc can be ejected before the background formatting is finished, in that case only one AVDP exists on the MRW disc. Note that at an early eject the drive must format all non-recorded areas up to the highest sector number recorded by

the host, this could cause a significant delay in the early eject process. Implementations are recommended to allocate the lowest numbered blocks available while background physical formatting is in progress.
The background physical formatting shall not influence the recording of the LVID. At early eject the LVID shall be recorded in the same way as it will be recorded on rewritable media that do not support background physical formatting.

OSTA
Optical Storage Technology Association

UDF Document Change Notice
DCN-5076

Document: OSTA Universal Disk Format DCN-5076
Subject: *Recommendations for incremental writing on DVD+R and DVD+RW (UDF 1.50 and higher)*
Date: June 17, 2002, editorial update March 12, 2003.

Description:
This DCN defines recommendations for incremental writing on DVD+R and DVD+RW media. This DCN is applied to UDF 1.50 and higher.

Change:
Add new appendix 6.x (will be 6.13):

6.x Recommendations for DVD+R and DVD+RW Media
DVD+R and DVD+RW Media require special consideration due to its nature. The following guidelines are established to ensure interchange.

6.x.1 Use of UDF for incremental writing on DVD+R media
ECMA 167 requires an Anchor Volume Descriptor Pointer (AVDP) at sector 256 and either *N* or (*N* - 256), where n is the last recorded Physical Address on the media. The file system may be in an intermediate state before closing and still be interchangeable, but not strictly in compliance with ECMA 167. In the intermediate state, only one AVDP exists. It should exist at sector 256 or, if not possible due to a reserved Fragment, it shall exist at sector 512. Before the second AVDP has been recorded, the file system is in an intermediate state and is not strictly in compliance with ECMA 167.

Implementations should place file system control structures into virtual space and file data into real space. Reader implementations may cache the entire VAT; the size of the VAT should be considered by any UDF originating software.

The VAT may be located by using READ TRACK INFORMATION command. See SCSI-3 Multi Media Commands.

6.x.1.1 Requirements

- An intermediate state is allowed on DVD+R media in which only one AVDP is recorded; this single AVDP shall be at sector 256 or sector 512 and according to the multisession rules below.

- The Logical Volume Integrity descriptor shall be recorded and the volume marked as open. Logical volume integrity can be verified by finding the VAT ICB at the last recorded Physical Address. If the VAT ICB is present, the volume is clean; otherwise it is dirty.

- The Partition Header descriptor, if recorded, shall specify no Unallocated Space Table, no Unallocated Space Bitmap, no Partition Integrity Table, no Freed Space Table, and no Freed Space Bitmap. The drive is capable of reporting free space directly, eliminating the need for a separate descriptor.

- Each surface shall contain 0 or 1 read only partitions, 0 or 1 write once partitions, and 0 or 1 virtual partitions. DVD+R media should contain 1 write once partition and 1 virtual partition.

6.x.1.2 "Bridge" formats

ISO 9660 requires a Primary Volume Descriptor (PVD) at sector 16. If an ISO 9660 file system is desired, it may contain references to the same files as those referenced by ECMA 167 structures, or reference a different set of files, or a combination of the two.

6.x.1.3 End of session data

A session is closed to enable reading by DVD-ROM drives. The last complete session on the disc shall conform completely to ECMA 167 and have two AVDPs recorded. This shall be accomplished by writing data according to End of session data table below.

End of session data

Count	Description
1	Anchor Volume Descriptor Pointer
255	Implementation specific. May contain user data, file system structures, and/or link areas.
1	VAT ICB.

The implementation specific data may contain repeated copies of the VAT and VAT ICB. Compatibility with drives that do not accurately report the location of the last sector will be enhanced. Implementations shall ensure that enough space is available to record the end of session data. Recording the end of session data brings a volume into compliance with ECMA 167.

6.x.1.4 Multisession in DVD+R

The Volume Recognition Sequence and Anchor Volume Descriptor Pointer locations are specified by ECMA 167 to be at a location relative to the beginning of the disc. The

beginning of a disc shall be determined from a base address *S* for the purposes of finding the VRS and AVDP.
'*S*' is the logical sector number of the first data sector in the last existent session of the volume.
'*N*' is the logical sector number of the last recorded data sector on a disc.

There shall be no more than one writable partition or session at one time, and this session shall be the last session on the disc.

6.x.1.4.1 Volume Recognition Sequence
The following descriptions are added to UDF (see also ECMA 167 Part 2) in order to handle a multisession disc.

- The volume recognition area of the UDF Bridge format shall be the part of the volume space starting at sector *S* + 16.
- The volume recognition space shall end in the Session in which it begins. As a result of this definition, the volume recognition area always exists in the last session of a disc.

6.x.1.4.2 Anchor Volume Descriptor Pointer
The Anchor Volume Descriptor Pointers (AVDP) shall be recorded at the following logical sector numbers: *S* + 256 and *N* - 256. The AVDP at sector *N* - 256 shall be recorded before closing a session; it may not be recorded while a session is open.

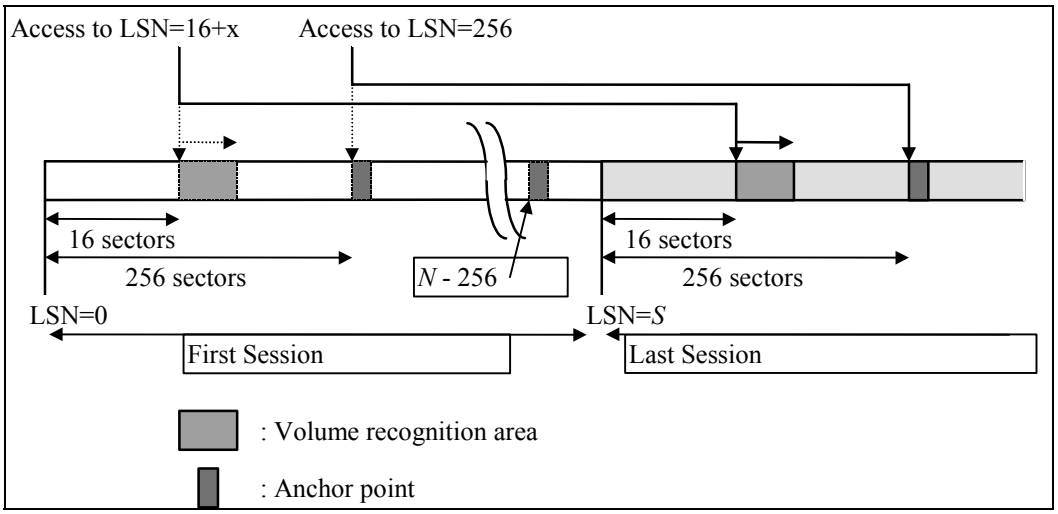
6.x.1.4.3 UDF Bridge format
The UDF Bridge format allows UDF to be added to a disc that may contain another file system. A UDF Bridge disc shall contain a UDF file system in its last session. The last session shall follow the rules described in "Multisession in DVD+R" section above. The disc may contain sessions that are based on ISO 9660, vendor unique, or a combination of file systems.

A new Main and Reserve Volume Descriptor Sequence may exist in each added session, and may be different than earlier VDSs.

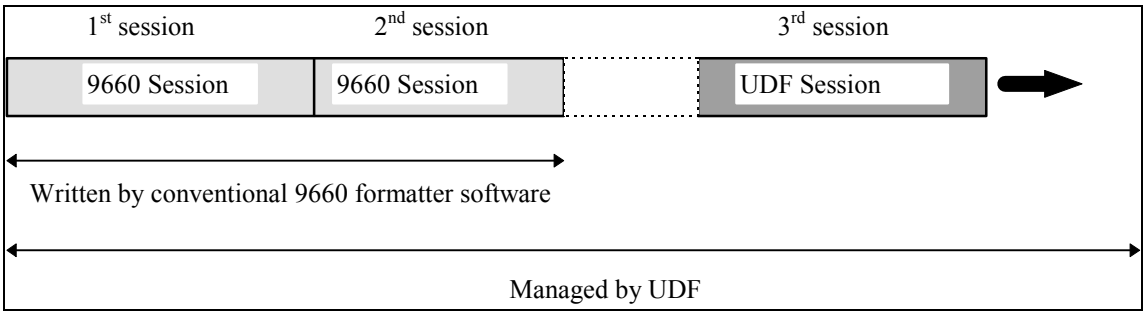
If the last session on a CD does not contain a valid UDF file system, the disc is not a UDF disc. Only the UDF structures in the last session, and any UDF structures and data referenced through them, are valid.

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. Some examples of UDF Bridge discs are shown below.

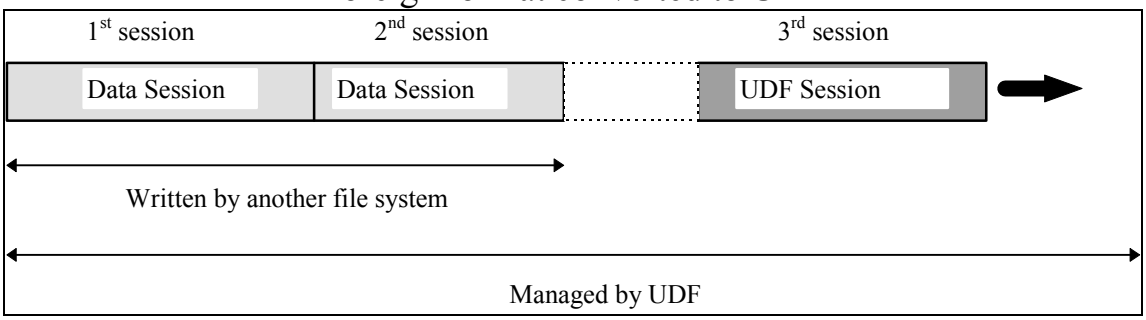
Multisession UDF disc



ISO 9660 converted to UDF



Foreign format converted to UDF



6.x.2 Use of UDF on DVD+RW 4.7 GBytes Basic Format media

DVD+RW 4.7 GBytes Basic Format media are random readable and writable, where needed the DVD+RW drive performs Read-Modify-Write cycles to accomplish this. For DVD+RW 4.7 GBytes Basic Format media the drive does not perform defect management. The DVD+RW 4.7 GBytes Basic Format provides the following features:

- A Physical Sector Size of 2048 Bytes
- 2048 Byte user data transfer
- Random read and write access
- Background physical formatting
- The Media Type is Overwritable (partition access type 4)

6.x.2.1 Requirements

- The packet length shall be set when the disc is formatted. The packet length shall be
  32 sectors (64 KB) if the disc is formatted as UDF 1.50 or UDF 2.00. It shall be 16 sectors (32 KB) if the disc is formatted as UDF 2.01 or later.
- Defective packets known at format time shall be allocated by the Non-Allocatable Space Stream (see 3.3.7.2).
- Sparing shall be managed by the host via the sparable partition and a sparing table.

NOTE: For UDF 1.50, a Non-Allocatable Space List as described in UDF 1.50 section 2.3.13 is used instead of the Non-Allocatable Space Stream as described in 3.3.7.2.

6.x.2.2 Background Physical Formatting

Physical formatting is performed by the drive in background. In implementing the host applications, the following requirements for the drive should be considered:

- After some minimal amount of formatting has been performed, the operation continues in background.

- At the initialization of the file system, after the Background Physical Formatting has been started, the host must record the first AVDP at sector 256. The second AVDP must be recorded after the Background physical Formatting has been finished. Before the second AVDP has been recorded, the file system is in an intermediate state and is not strictly in compliance with ECMA 167. The disc can be ejected before the background formatting has finished, and in that case only one AVDP exists. Note that at an early eject the drive must format all non-recorded areas up to the highest sector number recorded by the host, this could cause a significant delay in the early eject process. Implementations are recommended to allocate the lowest numbered blocks available while background physical formatting is in progress.

- The background physical formatting status shall not influence the recording of the LVID. At early eject the LVID shall be recorded in the same way as it will be recorded on rewritable media that do not support background physical formatting.

The physical formating may be followed by a verification pass. Defects found during the verification pass shall be enumerated in the *Non-Allocatable Space* list for UDF1.50 and the *Non-Allocatable Space* stream for UDF 2.00 and higher, see 3.3.7.2 (2.3.13 for UDF 1.50).

Finally, file system root structures shall be recorded. These mandatory file system and root structures include the Volume Recognition Sequence, the Anchor Volume Descriptor Pointers, the Volume Descriptor Sequences, a File Set Descriptor and a Root Directory.
Allocation for sparing shall occur during the formatting process. The sparing allocation may be zero in length.

The unallocated space descriptors shall be recorded and shall reflect the space allocated to not-spared defective areas and sector sparing areas.

The format may include all available space on the medium. However, formatting may be interrupted upon request by the user. Formatting may later be continued to the full space.

OSTA
Optical Storage
Technology Association

UDF Document Change Notice
DCN-5077

Document: OSTA Universal Disk Format DCN-5077
Subject: *Section 3.3.6 put out of order*
Date: September 26, 2002

Description:

This is a replacement for the disapproved DCN 5073. DCN 5073 tried to clarify the conversion of Extended Attributes to streams as mentioned in UDF section 3.3.6. In the FIC meetings of June 2001 and September 2002, the general opinion was that it was never intended to do 'automatic' conversion of any type of EA to a stream. Therefore, there is no use for section 3.3.6 any more and it was decided to replace 3.3.6 by a note, explaining that conversion is not allowed any more for the next UDF revision.
This DCN is for the next UDF revision after UDF 2.01 and as errata for UDF 2.00 and 2.01.

Change:

replace section 3.3.6 by:

3.3.6 Extended Attributes as named streams

NOTE: Because conversion of some types of Extended Attributes to a named stream appeared to be impossible and because it was never intended to allow automatic conversion of any EA to a named stream, this section is amended for UDF revisions after UDF 2.01. Conversion of any EA to a named stream is not allowed.

OSTA Optical Storage Technology Association

UDF Document Change Notice
DCN-5078

Document: OSTA Universal Disk Format DCN-5078
Subject: *UDF UniqueID clarifications*
Date: January 29, 2003. Editorial remarks March 7, 2003.

Description:

This DCN is for the next UDF revision and as errata for UDF 2.00 and 2.01.
Several UDF Unique ID issues are dealt with in this DCN. There is a relation with DCN-5063 about the UDF Unique ID Mapping Data in sections 3.3.7.1.*.
The issues dealt with here are:

- It is unclear what the UDF Unique ID value for a parent FID shall be.
- It is unclear whether or not a file or directory will get a new FID UDF Unique ID value during a rename operation.
- 3.3.3.4 contains a reference to ancient UDF 1.50 behavior.
- 3.3.5.1 contains an inaccurate duplication of what is already stated in 3.2.1.1.
- It is nowhere defined how the Next UniqueID value is defined in case of a file system using a VAT. It cannot be in the LVID.
- What are the real consequences of a 32-bit "wrap" as indicated in the first paragraph of 3.2.1.1? How must be dealt with the fact that after such a "wrap" the uniqueness of a 32-bits UDF UniqueID can no longer be guaranteed.

In a recent UDF reflector discussion (September/October 2002) it was concluded that the UniqueID value of an object shall not change in a rename or move operation, because the main reason for using UniqueIDs is to identify objects independent of the directory path and name of the object in the file system. Further the UniqueID usage for MacOs uses 32 bit UniqueID values instead of 31.

Change:

*Add to **3.2.1**:*

This structure is in the LVID Logical Volume Contents Use field.

*Replace the first paragraph of **3.2.1.1** by:*

3.2.1.1 Uint64 UniqueID

This field contains the Next *UniqueID* value to be used for the next new objects in the UDF UniqueID Mapping Data Stream, see 3.3.7.1. The Next *UniqueID* value is initialized to 16 because the value 0 is reserved for the root directory and system stream directory objects and the values 1-15 are reserved for use in Macintosh implementations. The Next *UniqueID* value monotonically increases with each assignment of a new UDF

UniqueID value for a newly created object as described below. Whenever the lower 32-bits of the Next *UniqueID* value reach #FFFFFFFF, the next increment is performed by incrementing the upper 32-bits by 1, as would be expected for a 64-bit value, but the lower 32-bits "wrap" to 16 (the initialization value). After such a "wrap", the uniqueness of a 32-bits FID UDF UniqueID value can no longer be guaranteed. Therefore the UDF UniqueID Mapping Data Stream shall be removed altogether if the value of Next *UniqueID* is higher than #FFFFFFFF.

In the 2nd paragraph of **3.2.1.1** *after:*

UniqueID is used whenever a new file or directory is created, or another name is linked to an existing file or directory.

add:

During a rename or move operation, the FID UniqueID value of an object shall not be changed and the values in the corresponding UDF Unique ID Mapping Entry shall remain consistent, see 3.3.7.1.2. The parent references of this mapping entry shall be updated when an object is moved to a different directory. When a FID is deleted, the mapping entry corresponding to the now unused UDF Unique ID shall not be re-used but be deleted or marked invalid.

At the end of the 2nd paragraph of **3.2.1.1** *after:*

… The same counts for File Entries/Extended File Entries used to define an Extended Attributes Space.

add:

A parent FID takes its Unique ID value from the 32 lower bits of the File Entry/Extended File Entry that is identified by the parent FID.

(Editorial note: Here, also text for system streams and system stream directory is added, see DCN-5072.)

At the very end of ***3.2.1.1*** *add:*

For file systems using a VAT, the function of the LVHD *UniqueID* field in the LVID is taken over by the VAT ICB File Entry UniqueID field with the addition that the first UniqueID value to be used for newly created objects will be the VAT ICB UniqueID value incremented once according to the incrementing policy described for Next *UniqueID* above in this section. In this way, no other object will have the same UniqueID value as the VAT File Entry.

*Replace the text in **3.3.3.4** by:*

Section 3.2.1 describes how the value for this field is set. For file systems using a VAT, the function of the LVHD UniqueID field in the LVID is taken over by the VAT File Entry UniqueID field, see 3.2.1.1.

NOTE: For UDF 2.00 and higher, the Unique ID value used in the UDF Unique ID Mapping Data is taken from the File Identifier Descriptor rather than from the File Entry.

*In **3.3.5.1** replace:*

The Unique ID field of Named Streams and Stream Directories shall be the same as the Unique ID of the main data stream.

by:

Section 3.2.1.1 describes how the Unique ID fields of File Identifier Descriptors and File Entries/Extended File Entries defining Named Streams and Stream Directories are set.

(Editorial remark: Text from DCN-5072 about system streams moved to 3.2.1.1)

OSTA Optical Storage Technology Association

UDF Document Change Notice
DCN-5079

Document: OSTA Universal Disk Format DCN-5079
Subject: *Clarify partition Access Type 3 and 4*
Date: September 27, 2002

Description:

In order to clarify the use of partition Access Type 3 and 4 for future media types a paragraph is added to the Partition Descriptor section.
This DCN is for the next UDF revision after UDF 2.01 and as clarification for previous revisions.

Change:

In 2.2.12 replace: Uint32 AccessType;
by: Uint32 **AccessType**;

add new section 2.2.12.2
and renumber current sections 2.2.12.2 thru 2.2.12.4:

2.2.12.2 Uint32 AccessType

For some rewritable/overwritable media types there may be confusion between partition access types 3 (rewritable) and 4 (overwritable).
Rewritable media are media that require some form of preprocessing before re-writing data (for example legacy MO). Such media shall have a Freed Space Bitmap or a Freed Space Table and shall use AccessType 3.
Overwritable media are media that do not require preprocessing before overwriting data (for example CD-RW). Such media shall not have a Freed Space Bitmap or a Freed Space Table and shall use AccessType 4.

OSTA Optical Storage Technology Association

UDF Document Change Notice
DCN-5081

Document: OSTA Universal Disk Format DCN-5081
Subject: *Icbtag Parent ICB Location issue.*
Date: Sep 10, 2002

Description:

The icbtag Parent ICB Location field is explained in ECMA 4/14.6.7 and UDF 2.3.5.3. It can easily be misunderstood, e.g. that there would be a relation with the parent directory, while this is definitely not the case. For UDF releases till 1.50 included (ECMA 2nd edition), it has only a meaning within the ICB hierarchy, which means that the value shall be zero for strategy 4. ECMA 3rd edition has a somewhat broader explanation, which adds more confusion, but there is still no relation with the parent directory. Further UDF 2.3.5.3 defines that the use of the field is optional, so a UDF implementation cannot rely on its presence. I think there is no much use in using this field, so I propose to change UDF 2.3.5.3.
See 3 OSTA UDF reflector postings on 29-03-2002 till 05-04-2002.
(If the field is also not used for strategy 4096, 2.3.5.3 can be further simplified).

Change:

In **2.3.5.3** *replace***:**

The use of this field is optional.

by:

For strategy 4 this field shall not be used and contain all zero bytes. For strategy type 4096 the use of this field is optional.

OSTA
Optical Storage
Technology Association

UDF Document Change Notice
DCN-5082

Document: OSTA Universal Disk Format DCN-5082
Subject: *Clarification of Volume Recognition Sequence*
Date: December 3, 2002

Description:

This DCN is for the next UDF revision after 2.01 and as errata for UDF 2.01.
From the "for read" text in 2.1.7 of UDF 2.01 it can be concluded that recording of a Volume Recognition Sequence in UDF 2.00 and lower revisions is not required because the meaning of "this" and "these" is not clear. What is meant however is that the requirement concerning the sector after the VRS is new in UDF 2.01. Further "block" should be replaced by "sector" and a NOTE about BOOT2 is added.

Change:

*In **2.1.7** replace:*

…
The block after the VRS shall be unrecorded or contain all #00.

by:

...
The first sector after the VRS shall be unrecorded or contain all #00 bytes.

*In **2.1.7** replace:*

for read: Implementers should expect that disks recorded by UDF 2.00 and earlier did not have this constraint, and should handle these cases accordingly.

by:

for read: Implementers should expect that media recorded by UDF 2.00 and lower revisions do not have the requirement mentioned above concerning the first sector after the VRS.

NOTE: Currently, no BOOT2 descriptor is defined for UDF, see 5.3.
Further, see ECMA part 2, 3/3.1, 3/3.2 and 3/9.1.

UDF Document Change Notice
DCN-5086

Document:	OSTA Universal Disk Format DCN-5086
Subject:	*Metadata Partition Map*
Date:	December 4th 2002 (last updated March 24th 2003)
Author:	Tom Jolly (Microsoft Corporation)

Description:

This proposal attempts to address three big problems with the current UDF formats as applied to rewritable media types.

- UDF includes no structures enabling integrity checking tools to locate all ICBs on a volume, other than brute force media scans or complete directory hierarchy walks. These can take an unacceptably large amount of time on a volume with many files. Most other file systems incorporate some means of locating ICBs (or their equivalent) typically in the form of 'inode tables / trees'. Additionally the UDF format does nothing to encourage allocation of ICBs in groups, thus guaranteeing more fragmentation, seeks, etc. Since seek times are relatively high for optical storage devices, some clustering of ICBs would benefit media scan time, and also improve performance in metadata intensive file system operations due to better cache memory utilization on the device.
- Defragmentation of UDF volumes is almost impossible while the file system is in use due to the fact that ICBs are referenced by LBN. So without fixing up all FIDs referencing an ICB, it cannot be moved. Finding all of these links is not a trivial task for volumes containing more than a few objects. The answer to this is virtualization of ICB addresses, so their block location can be changed without the need to locate & fix up all references.
- There is no duplication of metadata i.e. ICBs and directory information. This is becoming even more important with the advent of larger media (blue laser) which will be used in the AV space for storage of huge video files. This media is particularly sensitive to dust/dirt/damage due to the very high data density. It is unacceptable that this potentially irreplaceable content can be lost simply due to an unreadable/damaged root directory (for example). Typically the data file formats used in this area (MPEG) can sustain local dropout without widespread side effects.

Changes:

Add new row to table in section 2.1.5.2 "Entity Identifiers":

Metadata Partition Map	Partition Type Identifier	"*UDF Metadata Partition"	UDF Identifier Suffix

Add new row to table in section 6.1 "UDF Entity Identifier Definitions":

"*UDF Metadata Partition"	Describes UDF Metadata Partition

Add new row to table in section 6.2 "UDF Entity Identifier Values":

"*UDF Metadata Partition"	#2A, #55, #44, #46, #20, #4D, #65, #74, #61, #64, #61, #74, #61, #20, #50, #61, #72, #74, #69, #74, #69, #6F, #6E

Change sections 2.3.6 FE and – add second note:

NOTE: If a Metadata Partition Map is recorded in a volume then all *FileEntries*, allocation descriptor extents and directory data *shall* be recorded in the Metadata Partition – i.e. in logical blocks allocated to the *Metadata* and/or *Metadata Mirror* Files (see section 2.2.13 for details including exceptions).

Insert new section after 2.2.9 and renumber following sections.

2.2.10 Metadata Partition Map

This partition map *shall* be recorded for volumes that contain a single partition having an access type of 1 (read only) or 4 (overwritable). It *shall not* be recorded in all other cases.

See section 2.2.13 for further description of the metadata partition.

Layout of Type 2 partition map for metadata partition

RBP	Length	Name	Contents
0	1	Partition Map Type	Uint8 = 2
1	1	Partition Map Length	Uint8 = 64
2	2	Reserved	#00 bytes
4	32	Partition Type Identifier	EntityID
36	2	Volume Sequence Number	Uint16
38	2	Partition Number	Uint16
40	4	Metadata File Location	Uint32
44	4	Metadata Mirror File Location	Uint32
48	4	Metadata Bitmap Location	Uint32
52	4	Allocation Unit Size (blocks)	Uint32
56	2	Alignment Unit Size (blocks)	Uint16
58	1	Flags	Uint8
59	5	Reserved	#00 bytes

- Partition Type Identifier:
- Flags = 0
- Identifier = *UDF Metadata Partition
- IdentifierSuffix is recorded as in section **Error! Reference source not found.**.

- Partition Number = the number of this partition. Shall identify a Partition Descriptor associated with this partition. This shall match the partition number in the Type 1 map or Type 2 sparable map, one and only one of which *shall* also be recorded as appropriate to the media type.
- Metadata File Location = address of the block containing the File Entry for the metadata file. This address *shall* be interpreted as a logical block number within the physical or sparable partition associated with this partition map (see above "Partition Number" field).
- Metadata Mirror File Location = address of block containing the File Entry for the metadata file mirror. This address *shall* be interpreted as a logical block number within the physical or sparable partition associated with this partition map (see above "Partition Number" field).
- Metadata Bitmap Location = the address of the File Entry for the metadata bitmap file. This address *shall* reference a physical or sparable partition.
- Allocation Unit Size = the number of logical blocks per Allocation Unit for the metadata file (and mirror file) associated with this partition map. This value *shall* be an integer multiple of the larger of the following three values: (media ECC block size / logical block size); Packet Length (if a type 2 sparable partition map is recorded); 32.
- Alignment Unit Size (blocks) = all extents allocated to the Metadata File (or Mirror File) must have a starting Lbn which is an integer multiple of this value. This value shall be an integer multiple of the larger of the following: (media ECC block size / logical block size); Packet Length (if a type 2 sparable partition map is recorded).
- Flags:
  - Bit 0 – "Duplicate Metadata Flag": When set, indicates that the Metadata Mirror file has its own unique allocation (i.e. it duplicates the data in the Metadata File). When clear indicates that the Metadata Mirror File allocation descriptors describe the same allocation as the Metadata File allocation descriptors (i.e. the data is not duplicated, the data blocks are shared between both main and mirror files, but each File Entry and its associated allocation descriptors are unique and distinct).
  - Bits 1-7: Reserved. *Shall* be set to zero on write, and ignored on read.

NOTE: The Metadata Partition *shall* have an entry in the LVID Size and Free space tables (see 2.2.6).

NOTE: The Metadata File Location, Metadata Mirror File Location and Metadata Bitmap File Location Uint32 fields define File Entry locations. The number of blocks allocated for each File Entry shall be one logical block.

Insert second NOTE in section 2.3.4 (File Identifier Descriptor)

NOTE: On logical volumes where a metadata partition map is recorded, all directory and stream directory data *shall* be recorded in the metadata partition (see 2.2.10), however the data space of streams *shall* be recorded in physical space.

Insert heading before 2.3.10 paragraph 8:

NOTE: For volumes in which a Virtual Partition Map is recorded:

Insert new paragraphs after last paragraph of 2.3.10:

NOTE: For volumes in which a Metadata Partition Map is recorded:

- Allocation descriptors identifying directory or stream directory data *shall* identify metadata space.
- Allocation descriptors identifying file or stream data *shall* identify physical space.
- Allocation descriptors recorded in metadata space *shall* use SHORT_ADs when identifying extents also in metadata space.
- Allocation descriptors having an extent type of 3 (continuation) *shall* identify an extent in the same partition in which the type 3 descriptor itself is recorded.
- Descriptors recorded in metadata space *shall* have their metadata space logical block number recorded in their descriptor tag TagLocation field, if applicable.

Insert new section before "Partition Descriptor" which now becomes 2.2.14.

2.2.13 Metadata Partition

The files and policies defined in this section facilitate rapid location of all metadata in the volume, promote clustering of ICBs / directory information, and optionally facilitate duplication of all metadata. This will, in most cases, greatly speed file system repair operations by eliminating the need to perform an exhaustive media scan, or directory traversal, solely for the purpose of locating ICBs. The clustering of metadata will also significantly improve performance of metadata intensive implementation operations. When the metadata duplication option is chosen, file system robustness to media damage is increased, at some cost to performance.

When a Type 2 Metadata Partition map is recorded, the Metadata File, Metadata Mirror File and Metadata Bitmap File *shall* also be recorded and maintained.

The allocation descriptors of the Metadata Mirror File File Entry *shall* either:

- reference the same extents in the physical/sparable partition as referenced by the allocation descriptors of the Metadata File - in this case the "Duplicate Metadata Flag" in the Metadata Partition Map Flags field *shall not* be set.

OR

- reference different extents thus duplicating all metadata.- in this case the "Duplicate Metadata Flag" in the Metadata Partition Map Flags field *shall* be set.

The File Entries for the Metadata, Metadata Mirror and Metadata Bitmap files *shall not* be referenced by any structure other than the Metadata Partition Map and *shall* have a

link count of 0. These files, when present, *shall* be recorded in the physical/sparable partition referenced by the metadata partition map.

The metadata partition map (see 2.2.10) defines a partition space in which all metadata (FSD, ICBs, allocation descriptors, and directory data) *shall* be recorded, with the sole exception of the ICBs and data comprising the Metadata, Metadata Mirror, and Metadata Bitmap files as described above.

File Entries describing directories or stream directories *shall* use either “immediate” allocation (i.e. the data is embedded in the File Entry - see ECMA 4/14.6.8 flag bits 0-2) or SHORT_ADs to describe the data space of the directory, since this data resides in the metadata partition along with the File Entry itself.

File Entries describing any other type of file data (including streams) *shall* use either “immediate” allocation, or LONG_ADs which *shall* reference the physical or sparable partition referenced by the metadata partition, to describe the data space of the file.

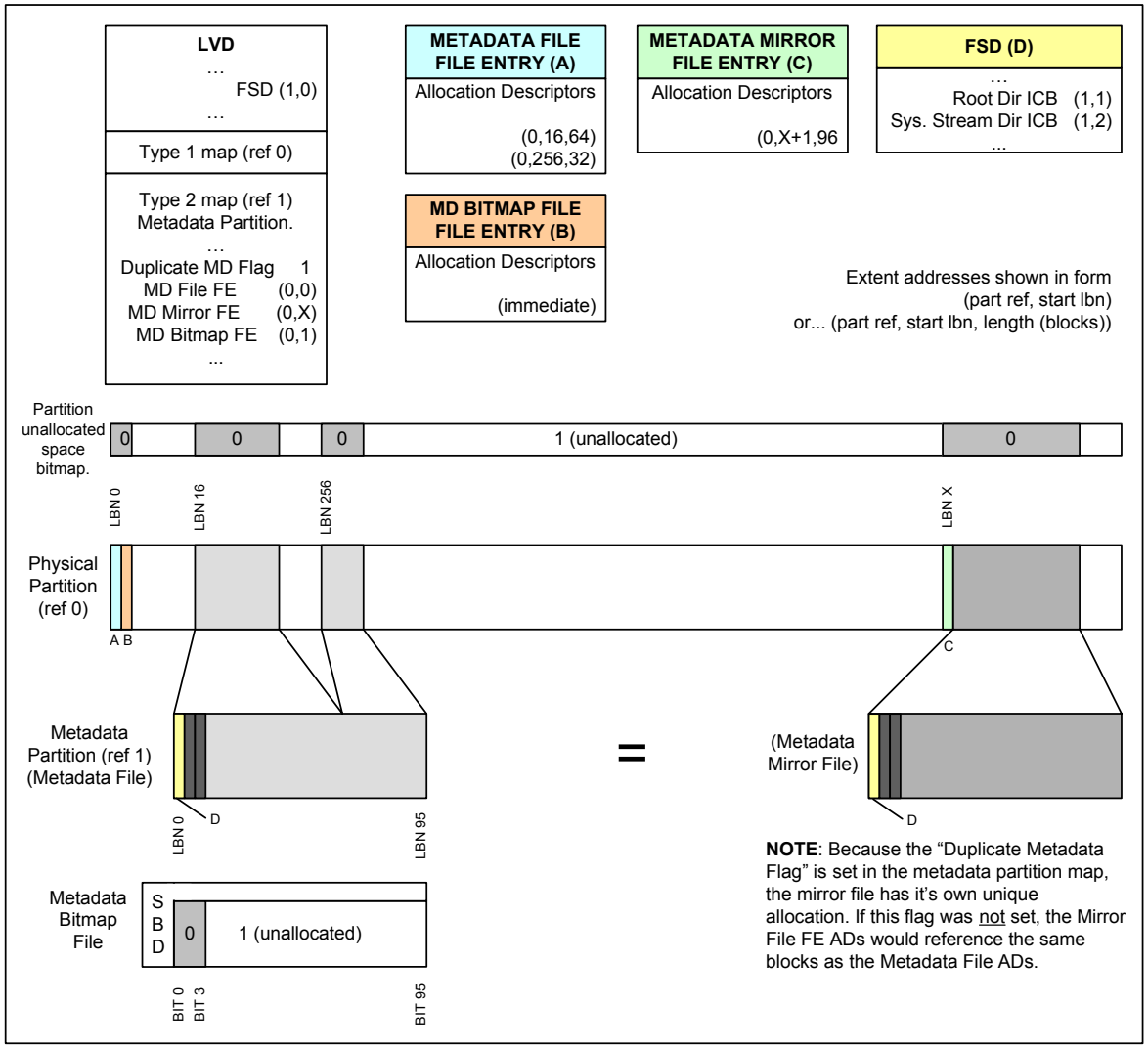
The “extent location” field of an allocation descriptor referencing data recorded in the metadata partition *shall* be interpreted as a block offset into the Metadata File. For example logical block 40 in the metadata partition corresponds to byte offset (40 * logical block size) in the Metadata File, which in turn (through the allocation descriptors for the metadata file) corresponds to some logical block in the associated physical/sparable partition.

Implementations *shall* support both the duplicate and shared allocation modes for the Metadata Mirror File (see above and 2.2.10, Metadata Partition Map, flags field). The file entry for the Metadata Mirror *shall* be actively maintained along with the Metadata File FE, but should be updated after the Metadata File FE.

If the “Duplicate Metadata Flag” is set in the Metadata Partition Map Flags, the Metadata Mirror File *shall* be maintained dynamically so that it contains identical data to the Metadata File at all times. In this case blocks in the metadata partition may be read from the same offset in either the Metadata Mirror file or the Metadata File. Data should be written first to the Metadata File and second to the Metadata Mirror File.

When the “Duplicate Metadata Flag” in the metadata partition map is set, implementations and repair utilities should consider the Metadata File content to be primary over that of the Metadata Mirror File. For example, a repair utility could repair the volume based on metadata read from the Metadata File (excepting unreadable portions which would be read from the Mirror) and then replace the contents of the Metadata Mirror file with that of the (now consistent) Metadata File.

Logical blocks allocated to the Metadata or Metadata Mirror Files *shall* be marked as allocated in the partition unallocated space bitmap, therefore a mechanism to determine available blocks within the metadata partition is needed. This is accomplished through the Metadata Bitmap file.



NOTE: the LBN values used in the diagram above are for illustrative purposes only and are not fixed. The partition references used are fixed as a consequence of the Metadata Partition implementation.

A more detailed description of these files and how they are used follows.

2.2.13.1 Metadata File (and Metadata Mirror File)

These files shall have file types of 250 (main) and 251 (Mirror) recorded in the IcbTag file type fields of their file entries. The UniqueID field of these file entries shall have a value of zero.

The allocation descriptors of these files *shall* at all times:

- Be SHORT_ADs (referencing space in the same physical/sparable partition in which the ICB resides).
- Either be of type "allocated and recorded" or type "not allocated".
- Have an extent length that is an integer multiple of the Allocation Unit Size specified in the Metadata Partition Map.
- Have a starting logical block number which is an integer multiple of the Alignment Unit Size specified in the Metadata Partition Map.
- The FE Information Length field *shall* be equal to (number of blocks described by the ADs for this stream * logical block size).

The allocation descriptors for this file *shall* describe only logical blocks which contain one of the below. No user data or other metadata types may be referenced.

- FSD
- ICB
- Extent of allocation descriptors.
- Directory or stream directory data (FIDs)
- An unused block marked free in the Metadata Bitmap File.

NOTE: In the case where the "Duplicate Metadata Flag" in the Metadata Partition Map is set, the allocations for the Metadata File and Metadata Mirror File should be as far apart (physically) as possible. Typically this is achieved by maximizing the difference between the start LBNs of the extents belonging to the file and its mirror. Likewise the file entries for these two files should be recorded as far apart as possible. Some drive/media combinations support "background physical formatting" or "incremental formatting" and implementations using such features should consider this when locating the metadata files and data. In such cases it may be practically impossible to position the files far apart without impacting the early eject time / media readability.

The Access Time and Modification time fields of the Metadata File and Mirror File ICBs *shall* be set to legal values at format time but need not be updated by a file system.

The ICBs for the Metadata File and Metadata Mirror file *shall* have null Stream_Directory_ICB and Extended_Attribute_ICB fields.

2.2.13.2 Metadata Bitmap File

This file *shall* have a file type of 252 recorded in the IcbTag File Type field of its file entry. The UniqueID field of this file entry shall have a value of zero.

This file contains a Space Bitmap Descriptor describing the utilization of blocks allocated to the Metadata File (i.e. this is a bitmap describing allocated space for the metadata partition). Bit zero of the bitmap corresponds to the first block in the aforementioned file, bit one to the second, and so on. This also applies to the Metadata Mirror File since contents of the two files are identical (regardless of the "duplicate metadata flag").

If a bit in this bitmap is set (one) then the corresponding blocks within the Metadata File and Metadata Mirror File are *available* for use by new metadata.

NOTE: When the "Duplicate metadata flag" in the Metadata Partition Map is not set, these blocks are one and the same, since the allocation descriptors for the Metadata Mirror file reference the same blocks as those of the Metadata File.

If a bit in this bitmap is clear (zero) then the corresponding blocks are not available for use – i.e. they are either in use, or fall within an unallocated region of the Metadata File.

Other requirements for the Metadata Bitmap File:

- The descriptor tag fields *DescriptorCRC* and *DescriptorCRCLength* for this SBD *shall* be set to zero.
- The allocation descriptors for the Metadata Bitmap File *shall not* include any allocation descriptors of type "not allocated".
- The Information Length field of the File Entry for this file *shall* equal the size of the SBD (NOTE: SBD size includes bitmap portion).
- There shall be one bit in the bitmap for every block in the metadata partition.
- The Access Time and Modification time fields of the Metadata Bitmap File ICB *shall* be set to legal values at format time but need *not* be updated by a file system.
- The Metadata Bitmap File ICB *shall* have null StreamDirectoryIcb (if extended FE) and ExtendedAttributeICB fields.
- The descriptor TagLocation field of this SBD *shall* be set to the logical block number of the first block allocated to the Metadata Bitmap File.

2.2.13.4 Procedure for allocating blocks for new metadata.

Search for a set (one) bit in the Metadata Bitmap file, and clear it. The corresponding block within the metadata partition (Metadata and Metadata Mirror (if duplicate mode) files) may then be used for the new data. If there are no set (one) bits then the Metadata File (and Mirror if duplicate) must be extended as described in section 2.2.13.6 below.

2.2.13.5 Procedure for de-allocating metadata blocks.

Set (to one) the bit(s) in the Metadata Bitmap file corresponding to the block number(s) of the data within the metadata partition that is being de-allocated.

2.2.13.6 Recommended procedure for extending the Metadata Partition

These changes should be written to the device before the new blocks are allocated for use by metadata. It would be undesirable for such changes to sit in an implementation's write cache for so long that new metadata assigned to the blocks being described by the changes was written to the media first.

1. Verify that there is enough space in the Metadata File and Metadata Mirror File allocation descriptor chains for a new allocation descriptor. If not then allocate a new allocation descriptor extent.
2. Verify that the Metadata Bitmap file allocation is large enough to extend the bitmap to describe the additional blocks added to the metadata file, and if not then allocate block(s) for the metadata bitmap file.
3. Allocate a new extent of blocks (for the Metadata File) observing the size and alignment requirements specified in 2.2.13.1.
4. If the "Duplicate Metadata Flag" in the Metadata Partition Map is set, allocate a second extent of blocks observing the size and alignment requirements specified in 2.2.13.1, ideally as far away as possible from the first allocation (for the Metadata Mirror File).
5. Add a new allocation descriptor to the Metadata File, or modify existing descriptors, to reference the first newly allocated extent. If the "Duplicate Metadata Flag" in the Metadata Partition Map is not set, modify the Metadata Mirror file ADs to reference the same extent.
6. If a second extent of blocks was allocated above, add to the Metadata Mirror File a new allocation descriptor, or modify existing ADs, to reference this second extent.
7. If the new extents were added at the end of the Metadata File then increase the FE Information Lengths for the Metadata File, and Mirror, to include the new blocks.
8. If the Metadata Bitmap file was extended, increase its FE Information Length field to include the bits describing the additional blocks allocated to the Metadata files.
9. Set (set to one) the bits in the Metadata Bitmap file which correspond to the extent just added to the Metadata file, to indicate the blocks are available for use by new metadata.

2.2.13.7 Recommended procedure for reclaiming space from the Metadata Partition

Blocks allocated to the Metadata File, and its mirror, *shall* only be returned to the volume in one of the following two ways:

- Truncation of the Metadata File and its mirror.
- Marking the AD(s) for a region of the Metadata file, and it's mirror, as sparse (not allocated) and setting the corresponding bits in the Metadata Bitmap file to zero, indicating these blocks are not available for use.

Any region to be removed *shall*:

- Currently contain no referenced metadata (i.e. all corresponding bits in the Metadata Bitmap file *shall* already be set (one)).
- Match the size/alignment restrictions laid down in section 2.2.13.1

In the truncation case (metadata partition being truncated):

1. Update the SBD in the Metadata Bitmap File to reduce the bitmap size.
2. Update the Metadata Bitmap file entry Information Length to reflect the decreased bitmap size.
3. Update the Metadata File, and mirror, file entry Information Length fields to 'remove' the region.
4. Mark the de-allocated blocks as available in the partition unallocated space bitmap.

In the mark sparse case (region in middle of metadata partition being removed):

1. Clear the corresponding bits in the Metadata Bitmap file to zero.
2. Generate sparse (not allocated) allocation descriptor(s) in the Metadata File (and its mirror) for the region being de-allocated.
3. Mark the de-allocated blocks as available in the partition unallocated space bitmap.

Add new sentence at end of 2.3.5.2:

….File types 250, 251 and 252 shall be used for the Metadata File, Metadata Mirror File and Metadata Bitmap File respectively. See section 2.2.13 for more details. File types 253 to 255 shall not be used.

UDF Document Change Notice
DCN-5089

Document:	OSTA Universal Disk Format DCN-5089
Subject:	*Partition Alignment & ECC Block Size Definition*
Date:	February 24th 2003 (updated March 3rd 2003)
Author:	Tom Jolly (Microsoft Corporation)

Description:

The term "ECC Block Size" is used in the UDF specification without any clarification.

Partition start block alignment currently is only specified for sparable partitions (for packet based media), when in fact it would be useful on some non packet media.

Changes:

Add new section to section 1.3.2 "Definitions":

ECC Block Size (bytes) This term refers to values defined in relevant device and/or media specifications. The reader should consult the appropriate document – for example, the "MMC" or "Mt. Fuji" specifications for C/DVD class media. For media exposing no such concept externally (e.g. hard disc) this term shall be interpreted to mean the sector size of the media.

Add new paragraph at end of section 2.2.12.2:

For a physical partition, the value of this field shall be an integral multiple of ("ECC Block Size" (divided by) sector size) for the media.

UDF Document Change Notice
DCN-5090

Document:	OSTA Universal Disk Format DCN-5090
Subject:	*Non-allocatable space stream clarifications.*
Date:	March 3rd 2003, updated March 20th 2003
Author:	Tom Jolly (Microsoft Corporation)

Description:

The sections describing use of the non-allocatable space stream and sparing table are unclear and contradictory.

Changes:

Replace section 3.3.7.2 first paragraph last sentence:

…."The *Non-Allocatable Space Stream* shall be recorded only on media systems that do not do defect management (e.g. CD-RW)"

With:

…."The *Non-Allocatable Space Stream* shall be recorded only on volumes with a sparable partition map recorded."

Replace section 3.3.7.2 paragraph 3 last sentence:

…."This list shall include both defective sectors found at format time and space allocated for sparing at format time."

With:

…."This stream shall include both defective packets found at format time and space allocated for sparing at format time."

Remove NOTE at end of 3.3.7.2.

Replace section 6.10.2.1 bullet 6:

- The host shall maintain a list of defects on the disc using a Non-Allocatable Space Stream (See 3.3.7.2).

With:

- Defective packets known at format time shall be allocated by the Non-Allocatable Space Stream (see 3.3.7.2).

In section 6.10.2.2 replace: Defects found during …

by: Defective packets found during …

Editorial note: An editorial consequence of "*Replace section 6.10.2.1 bullet 6:*" is that for DCN-5076 the following change must be made:

Replace section 6.x.2.1 bullet 2:

- The implementation shall maintain a Non-Allocatable Space Stream, see 3.3.7.2

with:

- Defective packets known at format time shall be allocated by the Non-Allocatable Space Stream (see 3.3.7.2).