[MS-DLTCS-Diff]:

Distributed Link Tracking Central Store Protocol

Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation ("this documentation") for protocols, file formats, data portability, computer languages, and standards support. Additionally, overview documents cover inter-protocol relationships and interactions.
- **Copyrights**. This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you can make copies of it in order to develop implementations of the technologies that are described in this documentation and can distribute portions of it in your implementations that use these technologies or in your documentation as necessary to properly document the implementation. You can also distribute in your implementation, with or without modification, any schemas, IDLs, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications documentation.
- **No Trade Secrets**. Microsoft does not claim any trade secret rights in this documentation.
- **Patents**. Microsoft has patents that might cover your implementations of the technologies described in the Open Specifications documentation. Neither this notice nor Microsoft's delivery of this documentation grants any licenses under those patents or any other Microsoft patents. However, a given Open Specifications document might be covered by the Microsoft Open Specifications Promise or the Microsoft Community Promise. If you would prefer a written license, or if the technologies described in this documentation are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting iplg@microsoft.com.
- **License Programs**. To see all of the protocols in scope under a specific license program and the associated patents, visit the Patent Map.
- **Trademarks**. The names of companies and products contained in this documentation might be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights. For a list of Microsoft trademarks, visit www.microsoft.com/trademarks.
- **Fictitious Names**. The example companies, organizations, products, domain names, email addresses, logos, people, places, and events that are depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

Reservation of Rights. All other rights are reserved, and this notice does not grant any rights other than as specifically described above, whether by implication, estoppel, or otherwise.

Tools. The Open Specifications documentation does not require the use of Microsoft programming tools or programming environments in order for you to develop an implementation. If you have access to Microsoft programming tools and environments, you are free to take advantage of them. Certain Open Specifications documents are intended for use in conjunction with publicly available standards specifications and network programming art and, as such, assume that the reader either is familiar with the aforementioned material or has immediate access to it.

Support. For questions and support, please contact dochelp@microsoft.com.

Revision Summary

Date	Revision History	Revision Class	Comments
3/2/2007	1.0	Major	Updated and revised the technical content.
4/3/2007	1.1	Minor	Clarified the meaning of the technical content.
5/11/2007	1.2	Minor	Addressed EU feedback
6/1/2007	1.3	Minor	Clarified the meaning of the technical content.
7/3/2007	1.3.1	Editorial	Changed language and formatting in the technical content.
8/10/2007	1.3.2	Editorial	Changed language and formatting in the technical content.
9/28/2007	1.3.3	Editorial	Changed language and formatting in the technical content.
10/23/2007	2.0	Major	Converted document to unified format.
1/25/2008	2.0.1	Editorial	Changed language and formatting in the technical content.
3/14/2008	2.0.2	Editorial	Changed language and formatting in the technical content.
6/20/2008	2.0.3	Editorial	Changed language and formatting in the technical content.
7/25/2008	2.0.4	Editorial	Changed language and formatting in the technical content.
8/29/2008	3.0	Major	Added a section.
10/24/2008	3.0.1	Editorial	Changed language and formatting in the technical content.
12/5/2008	4.0	Major	Updated and revised the technical content.
1/16/2009	4.0.1	Editorial	Changed language and formatting in the technical content.
2/27/2009	4.0.2	Editorial	Changed language and formatting in the technical content.
4/10/2009	4.0.3	Editorial	Changed language and formatting in the technical content.
5/22/2009	4.0.4	Editorial	Changed language and formatting in the technical content.
7/2/2009	4.0.5	Editorial	Changed language and formatting in the technical content.
8/14/2009	4.0.6	Editorial	Changed language and formatting in the technical content.
9/25/2009	4.1	Minor	Clarified the meaning of the technical content.
11/6/2009	4.1.1	Editorial	Changed language and formatting in the technical content.
12/18/2009	4.1.2	Editorial	Changed language and formatting in the technical content.
1/29/2010	4.2	Minor	Clarified the meaning of the technical content.
3/12/2010	4.2.1	Editorial	Changed language and formatting in the technical content.
4/23/2010	4.2.2	Editorial	Changed language and formatting in the technical content.
6/4/2010	4.2.3	Editorial	Changed language and formatting in the technical content.
7/16/2010	4.2.3	None	No changes to the meaning, language, or formatting of the technical content.

Date	Revision History	Revision Class	Comments
8/27/2010	4.2.3	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2010	4.2.3	None	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	4.2.3	None	No changes to the meaning, language, or formatting of the technical content.
1/7/2011	4.2.3	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2011	4.2.3	None	No changes to the meaning, language, or formatting of the technical content.
3/25/2011	4.2.3	None	No changes to the meaning, language, or formatting of the technical content.
5/6/2011	4.2.3	None	No changes to the meaning, language, or formatting of the technical content.
6/17/2011	4.3	Minor	Clarified the meaning of the technical content.
9/23/2011	4.3	None	No changes to the meaning, language, or formatting of the technical content.
12/16/2011	4.3	None	No changes to the meaning, language, or formatting of the technical content.
3/30/2012	4.3	None	No changes to the meaning, language, or formatting of the technical content.
7/12/2012	4.3	None	No changes to the meaning, language, or formatting of the technical content.
10/25/2012	4.3	None	No changes to the meaning, language, or formatting of the technical content.
1/31/2013	4.3	None	No changes to the meaning, language, or formatting of the technical content.
8/8/2013	4.3	None	No changes to the meaning, language, or formatting of the technical content.
11/14/2013	4.3	None	No changes to the meaning, language, or formatting of the technical content.
2/13/2014	4.3	None	No changes to the meaning, language, or formatting of the technical content.
5/15/2014	4.3	None	No changes to the meaning, language, or formatting of the technical content.
6/30/2015	4.3	None	No changes to the meaning, language, or formatting of the technical content.
7/14/2016	4.3	None	No changes to the meaning, language, or formatting of the technical content.
6/1/2017	4.3	None	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

This document specifies the Distributed Link Tracking: Central Store Protocol.

Distributed Link Tracking (DLT) refers to a set of protocols used to determine the new location of a file that has moved, whether the file has moved within a computer or between computers in a network that shares files with the Server Message Block (SMB) Protocol, as specified in [MS-SMB].

The Distributed Link Tracking: Central Store Protocol is used to store the tables of the Distributed Link Tracking: Central Manager Protocol, as specified in [MS-DLTM], and to transmit table updates between instances of the Distributed Link Tracking: Central Store Protocol servers.

This protocol is based on Active Directory, as specified in [MS-ADTS]. Specifically, this protocol treats Active Directory as a transport itself. For example, if a server of the Distributed Link Tracking: Central Manager Protocol writes an object to Active Directory according to the Distributed Link Tracking: Central Store Protocol, Active Directory replicates that object to another computer, where it can be read by another instance of a Distributed Link Tracking (DLT) Central Manager server. This Distributed Link Tracking: Central Store Protocol describes how Active Directory objects are defined, updated, and interpreted. The replication mechanism for Active Directory itself is as specified in [MS-ADTS].

In addition to the Distributed Link Tracking: Central Store Protocol, there are two other protocols that make up Distributed Link Tracking:

- The Distributed Link Tracking: Central Manager Protocol, as specified in [MS-DLTM], is a remote procedure call (RPC)-based protocol that is used to send information from protocol clients to servers about files that have been moved between computers or between volumes within a computer, information such as a unique ID for a file and the ID of the computer on which a file is currently located. This protocol is also used to query for the identity of the computer that currently holds a file.

- The Distributed Link Tracking: Workstation Protocol, as specified in [MS-DLTW], is an RPC-based protocol that is used to determine a file's current Universal Naming Convention (UNC) location. Clients of this protocol can use the Distributed Link Tracking: Central Manager Protocol to determine which computer currently holds a particular file. This allows the client to determine the correct instance of the DLT Workstation server to contact.

Sections 1.5, 1.8, 1.9, 2, and 3 of this specification are normative. All other sections and examples in this specification are informative.

1.1 Glossary

This document uses the following terms:

Active Directory: A general-purpose network directory service. Active Directory also refers to the Windows implementation of a directory service. Active Directory stores information about a variety of objects in the network. Importantly, user accounts, computer accounts, groups, and all related credential information used by the Windows implementation of Kerberos are stored in Active Directory. Active Directory is either deployed as Active Directory Domain Services (AD DS) or Active Directory Lightweight Directory Services (AD LDS). [MS-ADTS] describes both forms. For more information, see [MS-AUTHSOD] section 1.1.1.5.2, Lightweight Directory Access Protocol (LDAP) versions 2 and 3, Kerberos, and DNS.

CurrentRefreshTime: The current time, in units of days, measuring the time since the value was initialized.

distinguished name (DN): A name that uniquely identifies an object by using the relative distinguished name (RDN) for the object, and the names of container objects and domains that contain the object. The distinguished name (DN) identifies the object and its location in a tree.

domain: A set of users and computers sharing a common namespace and management infrastructure. At least one computer member of the set must act as a domain controller (DC) and host a member list that identifies all members of the domain, as well as optionally hosting the Active Directory service. The domain controller provides authentication (2) of members, creating a unit of trust for its members. Each domain has an identifier that is shared among its members. For more information, see [MS-AUTHSOD] section 1.1.1.5 and [MS-ADTS].

domain controller (DC): The service, running on a server, that implements Active Directory, or the server hosting this service. The service hosts the data store for objects and interoperates with other DCs to ensure that a local change to an object replicates correctly across all DCs. When Active Directory is operating as Active Directory Domain Services (AD DS), the DC contains full NC replicas of the configuration naming context (config NC), schema naming context (schema NC), and one of the domain NCs in its forest. If the AD DS DC is a global catalog server (GC server), it contains partial NC replicas of the remaining domain NCs in its forest. For more information, see [MS-AUTHSOD] section 1.1.1.5.2 and [MS-ADTS]. When Active Directory is operating as Active Directory Lightweight Directory Services (AD LDS), several AD LDS DCs can run on one server. When Active Directory is operating as AD DS, only one AD DS DC can run on one server. However, several AD LDS DCs can coexist with one AD DS DC on one server. The AD LDS DC contains full NC replicas of the config NC and the schema NC in its forest. The domain controller is the server side of Authentication Protocol Domain Support [MS-APDS].

FileId: The FileLocation of a file at the time it was originally created. A file's FileId never changes.

FileLocation: A VolumeID with an appended ObjectID, which together represent the location of a file at some point in time, though the file might no longer be there. FileLocation values are stored in droid (CDomainRelativeObjId) data structures.

FileTable: A table (with rows uniquely identified by a FileLocation or FileID) that contains the following fields: [PreviousFileLocation, FileLocation, FileID, RefreshTime]. For more information [MS-DLTM] see section 3.1.1. Maps a FileLocation or FileID to a current FileLocation.

flags: A set of values used to configure or report options or settings.

IntegerConvertedUnicodeString: A Unicode string created from a binary value. The string is a representation of the integer interpretation of the binary value. For example, a value of 0x10 would be represented as the string "16".

RefreshTime: The last time that information for an entry in the VolumeTable or FileTable has been refreshed by its VolumeOwner.

relative distinguished name (RDN): The name of an object relative to its parent. This is the leftmost attribute-value pair in the distinguished name (DN) of an object. For example, in the DN "cn=Peter Houston, ou=NTDEV, dc=microsoft, dc=com", the RDN is "cn=Peter Houston". For more information, see [RFC2251].

relative identifier (RID): The last item in the series of SubAuthority values in a security identifier (SID) [SIDD]. It distinguishes one account or group from all other accounts and groups in the domain. No two accounts or groups in any domain share the same RID.

ServerVolumeTable: A table (with rows uniquely identified by a VolumeID) that contains the following fields: [VolumeID, VolumeSequenceNumber, VolumeSecret, VolumeOwner, RefreshTime]. For more information see section 3.1.1.

StoreMaster: The single agent responsible for performing certain updates to file-link information stored in VolumeTable and FileTable within an Active Directory Table (ADT). For more information on VolumeTable and FileTable, see [MSDLT].

SystemObject: An object with Active Directory. This object is always at the distinguished name (DN) "CN=System,DC=DomainName", where DomainName is the name of the domain.

Universal Naming Convention (UNC): A string format that specifies the location of a resource. For more information, see [MS-DTYP] section 2.2.57.

VolumeID: A unique identifier that represents the identity of a file system volume.

VolumeOwner: A MachineID that is considered to be the owner of a VolumeID. A VolumeID can only have one VolumeOwner. For more information, see [MS-DLTM].

VolumeSecret: A value that is used to establish a VolumeOwner. For more information, see [MS-DLTM].

VolumeSequenceNumber: An integer value used to track the sequence of move notification messages received by the protocol server. See [MS-DLTM] section 2.2.2 for more information.

VolumeTable: Maps a VolumeID to a RefreshTime, VolumeSequenceNumber, VolumeSecret, and VolumeOwner. For more information, see [MS-DLTM].

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as defined in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2   References

Links to a document in the Microsoft Open Specifications library point to the correct section in the most recently published version of the referenced document. However, because individual documents in the library are not updated at the same time, the section numbers in the documents may not match. You can confirm the correct section numbering by checking the Errata.

1.2.1   Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information.

[MS-ADA1] Microsoft Corporation, "Active Directory Schema Attributes A-L".

[MS-ADA2] Microsoft Corporation, "Active Directory Schema Attributes M".

[MS-ADA3] Microsoft Corporation, "Active Directory Schema Attributes N-Z".

[MS-ADSC] Microsoft Corporation, "Active Directory Schema Classes".

[MS-ADTS] Microsoft Corporation, "Active Directory Technical Specification".

[MS-DLTM] Microsoft Corporation, "Distributed Link Tracking: Central Manager Protocol".

[MS-SMB] Microsoft Corporation, "Server Message Block (SMB) Protocol".

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.rfc-editor.org/rfc/rfc2119.txt

1.2.2   Informative References

[MS-DLTW] Microsoft Corporation, "Distributed Link Tracking: Workstation Protocol".

1.3   Overview

The Distributed Link Tracking: Central Store Protocol is designed to be used by the Distributed Link Tracking: Central Manager Protocol, as specified in [MS-DLTM]. The Distributed Link Tracking: Central

Store Protocol describes how to store the ServerVolumeTable and FileTable tables, as specified in [MS-DLTM] section 3.1.1, in Active Directory objects, where the DLT Central Manager servers are running on ALL domain controllers (DCs) of a domain. Because the Active Directory objects are replicated between servers, this allows updates by one DLT Central Manager server (as specified in [MS-DLTM]) to table entries to be communicated to other protocol server instances that are part of the same domain.<1>

As described in the introduction, DLT is composed of three protocols. The following is an example of the three protocols working together:

- A file is created on computer M1. M1 assigns identifiers to the file.

- Computer M0 takes note of the file, locally storing its identifiers.

- The file moves from computer M1 to M2, and from there to M3.

- Computer M0 finds the file in its new location in one of two ways:

    1. Using only the Distributed Link Tracking: Workstation Protocol:

        - M0 contacts M1, using the identifiers stored previously, and learns that the file was moved to M2.

        - M0 contacts M2, and learns that the file was moved to M3.

        - M0 contacts M3, and learns the file's new name and location.

    2. Using all three protocols:

        - M0 contacts a DLT Central Manager server to query the current location of the file.

        - The DLT Central Manager server queries its tables, which are stored by the Distributed Link Tracking: Central Store Protocol, and determines that the file is currently on computer M3.

        - M0 contacts the Distributed Link Tracking: Workstation Protocol on M3, and learns the file's new name and location.

1.4   Relationship to Other Protocols

The Distributed Link Tracking: Central Store Protocol is dependent on Active Directory, as specified in [MS-ADTS], which is the store for its tables.

The Distributed Link Tracking: Central Store Protocol imports conceptual tables used by the Distributed Link Tracking: Central Manager Protocol (as specified in [MS-DLTM] section 3.1.1) server implementation. It is used by a DLT Central Manager server to replicate information to other such servers.

1.5   Prerequisites/Preconditions

Agents using this protocol must have access to Active Directory, as specified in [MS-ADTS], and the right to access and modify objects within SystemObject.

1.6   Applicability Statement

The Distributed Link Tracking: Central Store Protocol is applicable in configurations in which the Distributed Link Tracking: Central Manager Protocol, as specified in [MS-DLTM], is being used.

1.7 Versioning and Capability Negotiation

There is no versioning or capability negotiation in this protocol.

1.8 Vendor-Extensible Fields

This protocol does not define any vendor-extensible fields.

1.9 Standards Assignments

Parameter	Value	Reference
VolumeTable object	"CN=VolumeTable,CN=FileLinks,CN=System"	As specified in [MS-ADTS] (see also section 2.2.2)
FileTable object	"CN=ObjectMoveTable,CN=FileLinks,CN=System"	As specified in [MS-ADTS] (see also section 2.2.4)

2 Messages

The following sections specify how Distributed Link Tracking: Central Store Protocol messages are transported, and their syntax.

2.1 Transport

Data MUST be transferred between agents of this protocol by replication of Active Directory, as specified in [MS-ADTS]. That is, this protocol treats Active Directory itself as a protocol. Writing objects into Active Directory causes the Active Directory replication mechanism to transfer those objects over the Active Directory underlying protocol where they can be read by another agent.

2.2 Message Syntax

This section defines the individual objects in Active Directory that make up this protocol. The full schema of these objects is as specified in [MS-ADTS].

2.2.1 FileLinks Object

The FileLinks object is the container object for all link information and is found at the following relative distinguished name (RDN) within SystemObject.

    CN=FileLinks

The schema definition for the FileLinks object is the **fileLinkTracking** object, as specified in [MS-ADSC] section 2.50.

Exactly one FileLinks object MUST exist.

2.2.2 VolumeTable Object

The VolumeTable object represents the ServerVolumeTable, as specified in [MS-DLTM] section 3.1.1, and MUST be stored in Active Directory as an object at the following RDN within SystemObject.

    CN=VolumeTable,CN=FileLinks

The schema definition for the VolumeTable object MUST be the **linkTrackVolumeTable** object, as specified in [MS-ADSC] section 2.82.

2.2.3 VolumeTableEntry Object

The VolumeTableEntry object represents an entry in the ServerVolumeTable, as specified in [MS-DLTM] section 3.1.1, and MUST be stored in the VolumeTable object (section 2.2.2).

The schema definition for the VolumeTableEntry object MUST be the **linkTrackVolEntry** object specified in [MS-ADSC] section 2.81, and has an RDN of the following form.

    VolumeTableRDN = "CN=" + VolumeIDString
    VolumeIDString = A VolumeID in the form of a HexConvertedUnicodeString

A sample VolumeTableEntry object distinguished name (DN) is shown in the following example where "DomainName" is the name of the domain.

```
CN=E3D954B2D0A711D08CB600C04FD90F85,
+ CN=VolumeTable,CN=FileLinks,CN=System,
+ DC=DomainName
```

There are two special VolumeTableEntry objects with specific RDNs in the VolumeTable object that are not used as in the preceding example:

- **CurrentRefreshTimeEntry**: This entry MUST have an RDN of "0000000000000000", as if its RDN indicated a VolumeID consisting of a string of all "0" characters. Only the **seqNotification** attribute of this entry is used. It MUST be a 32-bit unsigned integer value in the form of an IntegerConvertedUnicodeString. The **seqNotification** attribute MUST be set to the value of the CurrentRefreshTime. All other attributes MUST NOT be set and MUST be ignored.

  Updates to this value are specified in Timer Events (section 3.1.6).

- **FileTableCounterEntry**: This entry MUST have an RDN of "QT_Counter". Only the **linkTrackSecret** attribute of this entry SHOULD be used. All other attributes MUST NOT be set, and MUST be ignored. This is an 8-byte binary value, representing a partial count of the number of entries in the FileTable. Updates to this value are specified in Timers (section 3.1.2).

2.2.4 FileTable Object

The FileTable object represents the FileTable, as specified in [MS-DLTM] section 3.1.1, and MUST be stored in Active Directory as an object at the following RDN within SystemObject.

```
CN=ObjectMoveTable,CN=FileLinks
```

The schema definition for the FileTable object MUST be the **linkTrackObjectMoveTable** object specified in [MS-ADSC] section 2.79.

2.2.5 FileTableEntry Object

The FileTableEntry object represents an entry in the FileTable, as specified in [MS-DLTM] section 3.1.1, and MUST be stored in the FileTable object (section 2.2.4).

The schema definition for the FileTableEntry object MUST be the **linkTrackOMTEntry** object specified in [MS-ADSC] section 2.80, and MUST have an RDN of the following form.

```
FileTableRDN = "CN=" + VolumeIDString + ObjectIDString
VolumeIDString = VolumeID in the form of a HexConvertedUnicodeString
ObjectIDString = ObjectID in the form of a HexConvertedUnicodeString
```

A sample FileTableEntry object DN is shown in the following example, where "DomainName" is the name of the domain.

```
CN=E3D954B2D0A711D08CB600C04FD90F85
+ 6454147C5A29427F-B7B03982C5202C2A,
+ CN=ObjectMoveTable,CN=FileLinks,CN=System,
```

```
      + DC=DomainName
```

2.3   Directory Service Schema Elements

The Distributed Link Tracking: Central Store Protocol accesses the Directory Service schema classes and attributes listed in the following table. For the syntactic specifications of the following <Class> or <Class> <Attribute> pairs, refer to [MS-ADSC], [MS-ADA1], [MS-ADA2], and [MS-ADA3].

Class	Attribute
fileLinkTracking (section 2.50)	
linkTrackVolumeTable (section 2.82)	
linkTrackVolEntry (section 2.81)	seqNotification (section 2.239), linkTrackSecret (section 2.362), volTableIdxGUID (section 2.366), timeRefresh (section 2.308)
linkTrackObjectMoveTable (section 2.79)	
linkTrackOMTEntry (section 2.80)	currentLocation (section 2.137), birthLocation (section 2.84), oMTIndxGuid (section 2.51), timeRefresh (section 2.308)

3   Protocol Details

The Distributed Link Tracking: Central Store Protocol is used to store file link information in a VolumeTable table and a FileTable table within Active Directory. This protocol is used as a storage implementation by the DLT Central Manager server, as specified in [MS-DLTM], and is used to coordinate between instances of such servers.

3.1   Central Store Details

3.1.1   Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This protocol specification does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

The Distributed Link Tracking: Central Store Protocol maintains information about linked files. Information is in the form of two conceptual tables imported from (that is, shared with) the Distributed Link Tracking: Central Manager Protocol, as specified in [MS-DLTM]. Those tables are called ServerVolumeTable and FileTable, as specified in [MS-DLTM] and detailed in section 3.1.1. Hereafter in this document, the ServerVolumeTable is referred to simply as the VolumeTable.

The fields of VolumeTable MUST map to the **linkTrackVolEntry** object attributes, as specified in section 2.2.2, as follows.

Field	Attribute name
VolumeSequenceNumber	seqNotification
VolumeSecret	linkTrackSecret
VolumeOwner	volTableIdxGUID
RefreshTime	timeRefresh

The VolumeID of an entry in the VolumeTable MUST correspond to the RDN of the **linkTrackVolEntry** object, as described in section 2.2.3.

The fields of FileTable MUST map to the **linkTrackOMTEntry** object attributes, as specified in section 2.2.4, as follows:

Field	Attribute name
FileLocation	currentLocation
FileID	birthLocation
Flags	oMTIndxGuid
RefreshTime	timeRefresh

The **PreviousFileLocation** of an entry in the FileTable MUST correspond to the RDN of the **linkTrackOMTEntry**, as described in section 2.2.5.

The **Flags** field provides information on the state of an object, as follows.

										1										2										3	
0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1
0	0	U	D	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

U (Flags_Uncounted): This flag indicates that an entry in FileTable is not represented in the **FileTableCounterEntry**, as specified in section 2.2.3.

D (Flags_Deleted): This flag indicates that an entry in FileTable represents a file that has been deleted.

Some operations on the tables are performed by a single entity. This entity is called StoreMaster. The StoreMaster entity is the DC that is the relative identifier (RID) Master DC, as specified in [MS-ADTS].

3.1.2 Timers

TableMaintenanceTimer: Causes the StoreMaster agent to perform maintenance on VolumeTable, as specified in 2.2.2, and FileTable, as specified in 2.2.4. For the maintenance performed, see section 3.1.6. This is a recurring timer and MUST fire once per day.

3.1.3 Initialization

Implementations of this protocol MUST have access to the VolumeTable object and FileTable object. Access to Active Directory objects is specified in [MS-ADTS].

If the **CurrentRefreshTimeEntry** object specified in section 2.2.3 does not exist, it MUST be created with a CurrentRefreshTime value initially set to 0.

Note Because this protocol is used by servers that conform to the standards specified in [MS-DLTM], which run only on DCs, this protocol MUST only be accessed on DCs as well.

3.1.4 Higher-Layer Triggered Events

Aside from simply querying the entries in the VolumeTable object or FileTable object, agents using this protocol might need to perform more complex operations on the table. The following sections explain how to perform some of these operations.

3.1.4.1 An Agent Wants to Add an Entry to FileTable

In this scenario, the entry MUST be added to FileTable, according to the schema definition. In addition, the **Flags** field MUST be set to **Flags_Uncounted**.

3.1.4.2 An Agent Wants to Delete an Entry from FileTable

The entry in FileTable MUST NOT actually be deleted. Rather, the **Flags_Deleted** flag MUST be added to the **Flags** field for the entry in FileTable.

3.1.4.3 An Agent Wants to Know the Size of FileTable

Servers of the Distributed Link Tracking: Central Manager Protocol, as specified in [MS-DLTM] section 3.1.4.2, require information about the size of the FileTable table to determine whether the FileTable size limit has been reached during the processing of a MOVE_NOTIFICATION request.

This value MUST be calculated by incrementing the value of **FileTableCounterEntry**, as specified in section 2.2.3, once for each entry in the FileTable table that has the **Flags_Uncounted** flag set in the **Flags** field but that does not have the **Flags_Deleted** flag set.

Note This calculation does not modify the value of the **FileTableCounterEntry**.

3.1.5 Message Processing Events and Sequencing Rules

There are no messages to process in this protocol.

3.1.6 Timer Events

When the TableMaintenanceTimer timer expires, the StoreMaster agent MUST perform the following operations:

- Deleted files are cleaned up in the FileTable table:

  When a file is deleted, the **Flags_Deleted** flag is set in the **Flags** field of the corresponding FileTable entry, as specified in section 3.1.4. Any FileTable entry with this flag set MUST be deleted.

- Multiple entries representing a single chain are coalesced in the FileTable:

  - If multiple entries in the FileTable represent a single chain, they MUST be coalesced into a single FileTable entry. A chain of FileTable entries is a sequence of two or more consecutive entries that track a file through multiple moves, beginning with the file's original location in the first entry's **PreviousFileLocation** column, and ending in the file's most recent location in the last entry's **NextFileLocation** column.

  - The process of coalescing FileTable entries involves condensing a chain of FileTable entries into a single entry. The condensed single entry contains the **PreviousFileLocation** field that was originally specified in the first FileTable entry and the **NextFileLocation** field that was originally specified in the last FileTable entry.

  - Assume that FileTable contains two entries, and that the **NextFileLocation** field of the first entry matches the **PreviousFileLocation** field of the second entry. The second entry MUST be deleted from FileTable, and the first entry MUST be updated so that its **NextFileLocation** field has the value of the **NextFileLocation** field of the deleted entry.

3.1.7 Other Local Events

There are no additional local events.

4 Protocol Examples

Following is an example of the FileTable Coalescence, as specified in section 3.1.6. In this example, the file's original FileLocation entry (and therefore the file's FileID entry) is "A". Subsequently, the FileLocation entry changes to "B" and then to "C". FileTable then has the following entries.

PreviousFileLocation	NextFileLocation	FileID
A	B	(Unspecified)
B	C	A

Because these two entries represent a single file's move, they are consolidated into a single entry.

PreviousFileLocation	NextFileLocation	FileID
A	C	(Unspecified)

Note that where the **PreviousFileLocation** entry is the same as FileID, the **FileID** attribute is not stored in the entry, causing **FileID** to be inferred from the **PreviousFileLocation** value.

As a similar example, a file starts as A, moves to B, and then to C, and then to D. But in this example, the FileTable entry for the move from B to C has not been recorded yet. This causes the following entries to be in the FileTable, which cannot be coalesced.

PreviousFileLocation	NextFileLocation	FileID
A	B	(Unspecified)
C	D	A

5   Security

The following sections specify security considerations for implementers of the Distributed Link Tracking: Central Store Protocol.

5.1   Security Considerations for Implementers

There are no additional security considerations for implementers.

5.2   Index of Security Parameters

There are no security parameters associated with this protocol.

6   Appendix A: Product Behavior

The information in this specification is applicable to the following Microsoft products or supplemental software. References to product versions include released service packs.

- Windows 2000 Server operating system

- Windows Server 2003 operating system

Exceptions, if any, are noted below. If a service pack or Quick Fix Engineering (QFE) number appears with the product version, behavior changed in that service pack or QFE. The new behavior also applies to subsequent service packs of the product unless otherwise specified. If a product edition appears with the product version, behavior is different in that product edition.

Unless otherwise specified, any statement of optional behavior in this specification that is prescribed using the terms "SHOULD" or "SHOULD NOT" implies product behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term "MAY" implies that the product does not follow the prescription.

<1> Section 1.3: This protocol is implemented only on Windows 2000 Server and Windows Server 2003.

7   Change Tracking

No table of changes is available. The document is either new or has had no changes since its last release.

8 Index

A

Abstract data model 13
Adding entries 14
Applicability 8

C

Capability negotiation 9
Change tracking 19

D

Data model - abstract 13
Deleting entries 14
Directory service schema elements 12

E

Elements - directory service schema 12
Examples 16

F

Fields - vendor-extensible 9
FileLinks 10
FileLinks Object message 10
FileTable
   adding entries 14
   deleting entries 14
   overview 11
   size 14
FileTable Object message 11
FileTableEntry 11
FileTableEntry Object message 11

G

Glossary 5

H

Higher-layer triggered events 14

I

Implementer - security considerations 17
Implementers - security considerations 17
Index of security parameters 17
Informative references 7
Initialization 14
Introduction 5

L

Local events 15

M

Message processing 15
Messages