

[MS-DFSC]: Distributed File System (DFS): Referral Protocol Specification

Intellectual Property Rights Notice for Protocol Documentation

- This protocol 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 may make copies of it in order to develop implementations of the protocols, and may distribute portions of it in your implementations of the protocols or your documentation as necessary to properly document the implementation. This permission also applies to any documents that are referenced in the protocol documentation.
- Microsoft does not claim any trade secret rights in this documentation.
- Microsoft has patents that may cover your implementations of the protocols. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. If you are interested in obtaining a patent license, please contact protocol@microsoft.com.
- The names of companies and products contained in this documentation may be covered by trademarks or similar intellectual property rights. This notice does not grant any licenses under those rights.
- All other rights are reserved, and this notice does not grant any rights other than specifically described above, whether by implication, estoppel, or otherwise.

This protocol documentation is intended for use in conjunction with publicly available standard specifications, network programming art, and Microsoft Windows distributed systems concepts, and assumes that the reader either is familiar with the aforementioned material or has immediate access to it.

A protocol specification 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.

Revision Summary

Date	Revision History	Revision Class	Comments
03/02/2007	0.01		Version 1.0 release
06/01/2007	1.0	Major	Updated and revised the technical content.
07/03/2007	1.0.1	Editorial	Revised and edited the technical content.
07/20/2007	2.0	Major	Updated and revised the technical content.

Date	Revision History	Revision Class	Comments
08/10/2007	3.0	Major	Updated and revised the technical content.
09/28/2007	3.0.1	Editorial	Revised and edited the technical content.
10/23/2007	3.1	Minor	Updated the technical content.
11/30/2007	3.1.1	Editorial	Revised and edited the technical content.
01/25/2008	3.1.2	Editorial	Revised and edited the technical content.

Table of Contents

1	Introduction	5
1.1	Glossary	5
1.2	References	6
1.2.1	Normative References	6
1.2.2	Informative References.....	7
1.3	Protocol Overview (Synopsis).....	7
1.4	Relationship to Other Protocols.....	9
1.5	Prerequisites/Preconditions	9
1.5.1	Common Requirements	9
1.5.2	Client.....	9
1.5.3	DC or DFS Root Target Server	9
1.6	Applicability Statement	10
1.7	Versioning and Capability Negotiation.....	10
1.8	Vendor-Extensible Fields	10
1.9	Standards Assignments.....	10
2	Messages	11
2.1	Transport.....	11
2.2	Message Syntax	11
2.2.1	Common Conventions.....	12
2.2.1.1	Hostname	12
2.2.1.2	Sharename	12
2.2.1.3	AD DS Domain Name	12
2.2.1.4	UNC Path	12
2.2.1.5	DFS Root	13
2.2.1.6	DFS Link	13
2.2.1.7	DFS Root Target.....	13
2.2.1.8	DFS Link Target	13
2.2.1.9	DFS Target.....	13
2.2.2	SMB2_REQ_GET_DFS_REFERRAL	14
2.2.3	RESP_GET_DFS_REFERRAL	15
2.2.4	Referral Entry Types.....	16
2.2.4.1	DFS_REFERRAL_V1	16
2.2.4.2	DFS_REFERRAL_V2	17
2.2.4.3	DFS_REFERRAL_V3	18
2.2.4.3.1	DFS_REFERRAL_V3 (NameListReferral Bit Set to 0)	18
2.2.4.3.2	DFS_REFERRAL_V3 (NameListReferral Bit Set to 1)	20
2.2.4.4	DFS_REFERRAL_V4	21
3	Protocol Details	22
3.1	DFS Client Details	22
3.1.1	Abstract Data Model	29
3.1.2	Timers	30
3.1.3	Initialization	30
3.1.4	Higher-Layer Triggered Events.....	30
3.1.4.1	User/Application Initiated I/O Operation on a UNC Path	30
3.1.5	Message Processing Events and Sequencing Rules	33
3.1.5.1	I/O Operation to Target Fails with STATUS_PATH_NOT_COVERED	33
3.1.5.2	I/O Operation to Target Fails with an Error Other than STATUS_PATH_NOT_COVERED.....	34
3.1.5.3	I/O Operation to a DFS Root Target or DFS Link Target Succeeds	34
3.1.5.4	Receiving a Referral Response.....	34

3.1.5.4.1	Receiving a Domain Referral Response.....	34
3.1.5.4.2	Receiving a DC Referral Response	35
3.1.5.4.3	Receiving a Root Referral Response or Link Referral Response.....	35
3.1.5.4.4	Receiving a sysvol Referral Response.....	36
3.1.6	Timer Events.....	36
3.1.7	Other Local Events.....	36
3.2	DFS Root Target Server Details	37
3.2.1	Abstract Data Model.....	37
3.2.2	Timers	37
3.2.3	Initialization.....	37
3.2.4	Higher-Layer Triggered Events.....	37
3.2.5	Message Processing Events and Sequencing Rules	38
3.2.5.1	Receiving a DFS Referral Request	38
3.2.5.2	Receiving a Domain Referral Request	38
3.2.5.3	Receiving a DC Referral Request.....	38
3.2.5.4	Receiving a sysvol Referral Request	38
3.2.5.5	Receiving a Root Referral Request or Link Referral Request	38
3.2.6	Timer Events.....	42
3.2.7	Other Local Events.....	42
3.3	Domain Controller Details.....	42
3.3.1	Abstract Data Model.....	42
3.3.2	Timers	42
3.3.3	Initialization.....	42
3.3.4	Higher-Layer Triggered Events.....	42
3.3.5	Message Processing Events and Sequencing Rules	43
3.3.5.1	Receiving a DFS Referral Request	43
3.3.5.2	Receiving a Domain Referral Request	43
3.3.5.3	Receiving a DC Referral Request.....	44
3.3.5.4	Receiving a sysvol Referral Request	45
3.3.5.5	Receiving a Root Referral Request or Link Referral Request	47
3.3.6	Timer Events.....	47
3.3.7	Other Local Events.....	47
4	Protocol Examples	48
4.1	Domain Referral.....	48
4.2	DC Referral	49
4.3	Domain-Based DFS Root Referral.....	49
4.4	Domain-Based DFS Link Referral	50
4.5	Domain-Based DFS Root Referral Packet Trace	51
4.6	Standalone DFS Root Referral	53
5	Security	55
5.1	Security Considerations for Implementers.....	55
5.2	Index of Security Parameters.....	55
6	Appendix A: Windows Behavior	56
7	Index.....	62

1 Introduction

The Distributed File System (DFS): Referral Protocol allows file system clients to resolve names from a namespace distributed across many servers and geographies into local names on specific file servers. After names have been resolved, clients can directly access files on the identified servers by using file system protocols such as the [Server Message Block \(SMB\) Protocol](#) (as specified in [MS-SMB]), Network File System (NFS) (for more information, see [\[RFC3530\]](#)), and NCP (for more information, see [NOVELL]).

1.1 Glossary

The following terms are defined in [\[MS-GLOS\]](#):

8.3 Name
Active Directory (AD)
Active Directory Domain
Active Directory Domain Services (AD DS)
Anonymous User
Binary Large Object (BLOB)
DFS
DFS Client
DFS Client Target Failback
DFS Client Target Failover
DFS In-Site Referral Mode
DFS Link
DFS Link Target
DFS Metadata
DFS Namespace
DFS Namespace, Domain-Based
DFS Namespace, Standalone
DFS Path
DFS Referral
DFS Referral Site Costing
DFS Root
DFS Root Scalability Mode
DFS Root Target
Directory Object (or Object)
Distributed File System (DFS)
Domain
Domain Controller (DC)
NetBIOS Name
Primary Domain Controller (PDC)
Server Message Block (SMB)
Share
Unicode
Universal Naming Convention (UNC)

The following terms are specific to this document:

DFS Interlink: A **DFS link** whose target is another **DFS path**. The **DFS path** may be in the same **DFS namespace** as that of the **DFS interlink** or in another **DFS namespace**.

DFS Namespace Name: The second path component of a **DFS path**. In the **DFS path** \\MyDomain\\MyDfs\\MyDir, the DFS namespace name is MyDfs.

DFS Server: A server computer that runs the **DFS** service required to respond to **DFS referral** requests. Also interchangeably used to refer to the **DFS** service itself.

DFS Target: Either a **DFS root target** server or a **DFS link target** server.

Host Name: The name of a computer used for identification and access purposes by both humans and other computers in a network.

Share Name: The name of a **share**.

Site (or AD DS Site): A collection of one or more well-connected (reliable and fast) TCP/IP subnets, as specified in [\[MS-ADTS\]](#). For the purposes of this specification, only the case-sensitive nature of the site name is important.

Site Cost (or AD DS Site Cost): An **AD DS** administrator-defined numerical value meant to indicate the bandwidth or actual monetary cost of transmitting data between two **AD DS sites**. Only a comparison between two site cost values is meaningful, with a lower **AD DS site** preferred to a higher **AD DS site cost**.

Target Set: A **DFS referral** response consists of **DFS targets** that are grouped into sets based on various policies. One policy groups **DFS targets** into two sets: one consisting of all **DFS targets** in the same **AD DS site** as the **DFS client**, and the other consisting of all remaining **DFS targets**. Another policy creates **DFS target sets**, where each set contains **DFS targets** that have the same **AD DS site cost** from the **DFS client's** perspective, in increasing order of cost. By assigning non-default priorities to **DFS targets**, other policies are possible. For more information, see [\[MSDFS\]](#).

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

1.2 References

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. Please check the archive site, <http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624>, as an additional source.

[CIFS] Leach, P. and Naik, D., "A Common Internet File System (CIFS/1.0) Protocol", March 1997, http://www.microsoft.com/about/legal/intellectualproperty/protocols/BSTD/CIFS/dr_aft-leach-cifs-v1-spec-02.txt

If you have any trouble finding [CIFS], please check [here](#).

[MS-ADTS] Microsoft Corporation, "[Active Directory Technical Specification](#)", June 2007.

[MS-DFSNM] Microsoft Corporation, "[Distributed File System \(DFS\): Namespace Management Protocol Specification](#)", September 2007.

[MS-ERREF] Microsoft Corporation, "[Windows Error Codes](#)", January 2007.

[MS-FSCC] Microsoft Corporation, "[File System Control Codes](#)", July 2007.

[MS-GLOS] Microsoft Corporation, "[Windows Protocols Master Glossary](#)", March 2007.

[MS-SECO] Microsoft Corporation, "[Windows Security Overview](#)", January 2007.

[MS-SMB] Microsoft Corporation, "[Server Message Block \(SMB\) Protocol Specification](#)", July 2007.

[MS-SMB2] Microsoft Corporation, "[Server Message Block \(SMB\) Version 2.0 Protocol Specification](#)", July 2007.

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

[RFC2251] Wahl, M., Howes, T., and Kille, S., "Lightweight Directory Access Protocol (v3)", RFC 2251, December 1997, <http://www.ietf.org/rfc/rfc2251.txt>

[UNICODE] The Unicode Consortium, "Unicode Home Page", 2006, <http://www.unicode.org/>

1.2.2 Informative References

[MSDN-FILE] Microsoft Corporation, "Naming a File", <http://msdn2.microsoft.com/en-us/library/aa365247.aspx>

[MSDFS] Microsoft Corporation, "How DFS Works", March 2003, <http://technet2.microsoft.com/WindowsServer/en/library/a9096e88-1634-4da6-b820-537341d349061033.mspx>

[MSFSMT] Microsoft Corporation, "Microsoft File Server Migration Toolkit", September 2005, <http://www.microsoft.com/windowsserver2003/upgrading/nt4/tooldocs/msfsc.mspx>

[NOVELL] Chappell, L.A. and Hakes, D.E., "Novell's Guide to NetWare LAN Analysis, 2nd Edition", Novell Press, June 1994, ISBN: 0782113621.

[RFC1001] Network Working Group, "Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Concepts and Methods", RFC 1001, March 1987, <http://www.ietf.org/rfc/rfc1001.txt>

[RFC1002] Network Working Group, "Protocol Standard for a NetBIOS Service on a TCP/UDP Transport: Detailed Specifications", RFC 1002, March 1987, <http://www.ietf.org/rfc/rfc1002.txt>

[RFC1034] Mockapetris, P., "Domain Names—Concepts and Facilities", RFC 1034, November 1987, <http://www.ietf.org/rfc/rfc1034.txt>

[RFC2165] Veizades, J., Guttman, E., Perkins, C., and Kaplan, S., "Service Location Protocol", RFC 2165, June 1997, <http://www.ietf.org/rfc/rfc2165.txt>

[RFC3530] Shepler, S., et al., "Network File System (NFS) version 4 Protocol", RFC 3530, April 2003, <http://www.ietf.org/rfc/rfc3530.txt>

[SNIA] Storage Networking Industry Association, "Common Internet File System (CIFS) Technical Reference, Revision 1.0", March 2002, http://www.snia.org/tech_activities/CIFS/CIFS-TR-1p00_FINAL.pdf

1.3 Protocol Overview (Synopsis)

The Distributed File System (DFS) enables file system clients to access remote file **shares** by using a **DFS path**, (or virtual name) for the share, which is then transparently resolved to an actual **share name** on an actual file server. There is quite a bit of flexibility in this mapping. For example, **DFS** can enable the grouping of a set of shares located on different file servers into a unified namespace. DFS name resolution can also be used to transparently recover from a file server failure and to resolve a share to a geographically optimal file server for the specific client that is requesting

access. Without DFS, users of a network file system, such as **Server Message Block (SMB)**, need to know the names of all file servers, and shares that reside on those file servers, for which they require access. With DFS, users can navigate a unified namespace to access files and folders without knowledge of the names of individual file servers and shares that host the data. For more information about DFS, including DFS terminology, usage models, and the Microsoft DFS implementation, see [\[MSDFS\]](#).

After the DFS path has been resolved to an actual path, clients can directly access files on the identified servers by using standard remote file system protocols, such as the [Server Message Block \(SMB\) Protocol](#) (as specified in [MS-SMB]), [Server Message Block \(SMB\) Version 2.0 Protocol](#) (as specified in [MS-SMB2]), Network File System (NFS) (as specified in [\[RFC3530\]](#)), and NCP, as specified in [NOVELL]. When the client accesses the files and directories on the resolved share, DFS also provides a mechanism to cause the **DFS client** to perform a **DFS referral** request when accessing a link within the share.

The DFS: Referral Protocol supports two types of namespaces: **domain-based namespaces**, which offer high availability and load balancing, and stand-alone namespaces, which reside on a single **DFS root target** server and do not require **domain** infrastructure. In **stand-alone DFS namespaces**, clients issue root referral requests and link referral requests directly to the DFS root target server.

For domain-based namespaces, clients issue DFS referral requests to **domain controllers (DCs)** to discover the existence of **Active Directory Domain Services (AD DS)** domains and the existence of **DFS namespaces**. Clients issue referral requests to DCs in order to discover the DFS root target servers hosting specific DFS namespaces. Clients can also issue referral requests to DFS root target servers to discover other DFS root target servers that host a DFS namespace. Clients issue referral requests to DFS root target servers to discover the locations of **DFS link targets** (shares on file servers). After the components of a DFS path have been resolved to specific targets, clients then issue file system requests directly to file servers, using the appropriate remote file system protocol for that server.

In crossing DFS namespaces, clients may attempt to issue file system requests against **DFS links**. File servers can notify the client of the need for name resolution by using designated error messages in file system protocols. Clients then issue referral requests to resolve DFS links into DFS link targets. With newly resolved paths, clients then reissue file system requests directly to the appropriate file servers. The DFS links in a DFS namespace can also include paths that point to other DFS namespaces. A DFS link that points to another DFS namespace is called a **DFS interlink**.

The DFS: Referral Protocol is a command/acknowledge protocol that sends out a sequence of referral requests to eventually resolve the DFS path to an actual path. There are five types of referral requests, as specified in section [2.2.2](#):

- Domain referrals, which identify the AD DS domains in the AD DS forest to which the DFS client is joined and the AD DS domains in other forests, which are part of a trust relationship with the DFS client's forest. A tailored set of domain referrals unique to each client may be provided.
- DC referrals, which identify the DCs of a specific AD DS domain.
- Root referrals, which identify the DFS root targets of a specific DFS namespace.
- Link referrals, which identify the DFS link targets of a specific link in a DFS namespace.
- Sysvol referrals, which identify the DCs that host a domain's SYSVOL or NETLOGON shares.

Domain-joined clients issue all five types of referral requests, while non-domain-joined clients issue only **DFS root** and DFS link referral requests. Optionally, clients can also be used to administer DFS namespaces (see [\[MS-ADTS\]](#)).

Clients can maintain local caches of information that are received through referral requests to avoid future referral requests and to improve the performance of DFS resource access, as specified in section [3.1.1](#).

1.4 Relationship to Other Protocols

The DFS: Referral Protocol relies on the [Server Message Block \(SMB\) Protocol](#), (as specified in [MS-SMB]) or the [Server Message Block \(SMB\) Version 2.0 Protocol](#) (as specified in [MS-SMB2]) as its transport layer.

The DFS topology and configuration of DFS namespaces named by referral requests are maintained by the [Distributed File System \(DFS\): Namespace Management Protocol](#) (as specified in [MS-DFSNM]).

The DFS: Referral Protocol allows SMB (as specified in [MS-SMB] and [MS-SMB2]) file system clients to resolve names from a namespace distributed across many servers and geographies into local names on specific file servers. After names have been resolved, clients can directly access files on the identified servers by using file system protocols, such as the SMB Protocol (as specified in [MS-SMB]), SMB Version 2.0 Protocol (as specified in [MS-SMB2]), NFS (as specified in [RFC3530](#)), and NCP (as specified in [NOVELL]).

All protocols that use SMB to access files on remote machines depend on DFS if those files are located on shares in a DFS namespace.

Note The Common Internet File System (CIFS) Protocol Specification for the DFS: Referral Protocol (as specified in [\[CIFS\]](#) sections 2.9, 3.5, and 4.4.1) does not fully define the semantics of DFS, thus a CIFS implementation of DFS and an implementation that is compliant with this specification may not interoperate.

1.5 Prerequisites/Preconditions

1.5.1 Common Requirements

The SMB DFS referral request uses the TRANS2_GET_DFS_REFERRAL SMB message, as specified in [\[MS-SMB\]](#). This is available only in SMB versions "NT LM 0.12" and later. The SMB2 DFS referral request uses the FSCTL_DFS_GET_REFERRALS IOCTL, as specified in [\[MS-SMB2\]](#).

The DFS: Referral Protocol assumes that the DFS namespace, which is accessed by the DFS client and the **DFS metadata** that is used by the DFS: Referral Protocol, has already been set up—possibly through the [Distributed File System \(DFS\): Namespace Management Protocol](#), as specified in [MS-DFSNM]. It also assumes that the SYSVOL/NETLOGON shares that are located through sysvol referrals are already set up.

1.5.2 Client

DFS clients that issue referral requests may or may not be joined to an AD DS domain. DFS namespaces accessed by the DFS client can be domain-based DFS namespaces or stand-alone DFS namespaces.

1.5.3 DC or DFS Root Target Server

To permit the **DFS targets** in the referral response to be ordered on the basis of **AD DS site cost** between the client **site** and a DFS target site, a server joined to an AD DS domain MUST be able to determine, from AD DS, the AD DS site of a client from the client's IP address. If the server intends

to support **DFS referral site costing**, it MUST be able to perform the site cost queries between the client and the DFS targets, as specified in [\[MS-ADTS\]](#).

The server MUST be able to receive and process referral requests from **anonymous users**.

The **DFS server** uses the DFS metadata of a DFS namespace to process referral requests. For domain-based DFS namespaces, the server MUST be able to access the DFS metadata stored in the AD DS **object** of the DFS namespace. Location and format of the DFS metadata is specified in [\[MS-DFSNM\]](#) section 2.3.

1.6 Applicability Statement

The DFS: Referral Protocol is applicable for scenarios in which files that are stored across one or more shares— possibly including one or more file servers—and the share administrator would like to provide one or more of the following capabilities.

- A unified namespace across multiple shares, such that from a client's perspective there is no need to know about individual share namespaces.
- A fault-tolerant namespace, such that failure of a share causes the client to fail over to an alternate share.
- A geographically optimized namespace, such that the client will automatically access the share that contains the geographically "lowest cost" copy of the files.
- A share name that is logically independent of the file server or share name.

1.7 Versioning and Capability Negotiation

A client specifies the highest DFS referral version supported in the **MaxReferralLevel** field of the [SMB2_REQ_GET_DFS_REFERRAL \(section 2.2.2\)](#) message.

A server responds with the highest DFS referral version understood by both the client and the server in the **VersionNumber** field of each referral entry, as specified in section [2.2.4](#).

1.8 Vendor-Extensible Fields

There are no vendor-extensible fields defined in the DFS: Referral Protocol.

This protocol uses NTSTATUS values taken from the NTSTATUS number space, as specified in [\[MS-ERREF\]](#). Vendors SHOULD [<1>](#) reuse those values with their indicated meaning. Choosing any other value runs the risk of a collision in the future.

1.9 Standards Assignments

The DFS: Referral Protocol uses standards assignments as specified in [\[MS-SMB\]](#) section 2.1, as well as the following:

Parameter	Value	Reference
STATUS_PATH_NOT_COVERED	0xC0000257	[MS-ERREF]

2 Messages

The following sections specify how (DFS): Referral Protocol messages are transported and DFS: Referral Protocol message syntax.

2.1 Transport

The DFS: Referral Protocol uses either the Server Message Block (SMB) Protocol (as specified in [\[MS-SMB\]](#)) or Server Message Block (SMB) Version 2.0 Protocol (as specified in [\[MS-SMB2\]](#)) as its transport protocol.

When the DFS: Referral Protocol uses the SMB Protocol (as specified in [\[MS-SMB\]](#)) as the transport, the DFS: Referral Protocol MUST use a TRANS2_GET_DFS_REFERRAL request within an SMB_COM_TRANSACTION2 message to request a referral, and a TRANS2_GET_DFS_REFERRAL response within an SMB_COM_TRANSACTION2 to respond with referral information. The TRANS2_GET_DFS_REFERRAL request and response messages are specified in [\[MS-SMB\]](#) sections [2.2.13.17](#) and [2.2.13.18](#). The SMB_COM_TRANSACTION2 message is specified in [\[MS-SMB\]](#) section 2.2.13.

The underlying SMB protocol implementation MUST support the DFS flags in SMB packets as specified in [\[MS-SMB\]](#). The assignments used by the SMB Protocol that are relevant to the DFS: Referral Protocol are as follows:

Parameter	Value	Reference
TRANS2_GET_DFS_REFERRAL subcommand type for the SMB_COM_TRANSACTION2 packet.	0x0010	[MS-SMB] section 2.2.13 and 2.2.13.17
CAP_DFS in the SMB_COM_NEGOTIATE Server Response extension.	0x00001000	[MS-SMB] section 2.2.3
SMB_SHARE_IS_IN_DFS bit in the OptionalSupport field of the SMB_COM_TREE_CONNECT_ANDX response.	0x0002	[MS-SMB] section 2.2.7
SMB_FLAGS2_DFS in the Flags2 field of the SMB header.	0x1000	[MS-SMB] section 2.2.1

When the DFS: Referral Protocol uses the SMB Version 2.0 Protocol as the transport, the DFS: Referral Protocol client provides an opaque buffer to the SMB Version 2.0 Protocol that MUST be transferred within the REQ_GET_DFS_REFERRAL structure sent in an SMB2 IOCTL request message to request a referral. The SMB2 DFS referral request uses the FSCTL_DFS_GET_REFERRALS IOCTL, as specified in [\[MS-SMB2\]](#). To respond to a referral request, the DFS: Referral Protocol server provides an opaque buffer to the SMB Version 2.0 Protocol as a response, and the SMB Version 2.0 Protocol implementation MUST transfer it using the IOCTL response message. The SMB2 IOCTL request message is specified in [\[MS-SMB2\]](#) section 2.2.31 and the SMB2 IOCTL response message is specified in [\[MS-SMB2\]](#) section 2.2.32.

2.2 Message Syntax

This section describes the contents of the two SMB messages used by the DFS: Referral Protocol. These messages are [SMB2_REQ_GET_DFS_REFERRAL \(section 2.2.2\)](#) and [RESP_GET_DFS_REFERRAL \(section 2.2.3\)](#).

2.2.1 Common Conventions

All strings in [SMB2_REQ_GET_DFS_REFERRAL \(section 2.2.2\)](#) and [RESP_GET_DFS_REFERRAL \(section 2.2.3\)](#) messages MUST be encoded as null-terminated strings of UTF-16 characters, as specified in [\[UNICODE\]](#). Constructs of the form "<value>" in strings are placeholders to be replaced with client or server-specified values. For example, the string description "\\<domain>\<namespace>" would take the form "\\contoso\templates" when populated with the values "contoso" for the <domain> placeholder and "templates" for the <namespace> placeholder.

All paths in SMB2_REQ_GET_DFS_REFERRAL and RESP_GET_DFS_REFERRAL messages MUST be encoded with exactly one leading backslash, not two leading backslashes as is common to user-visible **UNC** paths. For example, the UNC path "\\server\namespace\directory\subdirectory\file" would be encoded as "\server\namespace\directory\subdirectory\file".

All 16-bit and 32-bit integers in SMB2_REQ_GET_DFS_REFERRAL and RESP_GET_DFS_REFERRAL messages MUST be encoded as unsigned little-endian values.

2.2.1.1 Hostname

Unless specified otherwise, a **hostname** MUST be a null-terminated **Unicode** character string, which MAY be a **NetBIOS name**, a DNS name (for more information, see [\[RFC1034\]](#)) or other name formats supported by name resolution mechanisms like the Service Location Protocol (for more information, see [\[RFC2165\]](#)).

2.2.1.2 Sharename

Unless specified otherwise, a sharename MUST be a null-terminated Unicode character string whose format is dependent on the underlying file server protocol that is used to access the share. Examples of file server protocols are the SMB Protocol (as specified in [\[MS-SMB\]](#)), NFS (for more information, see [\[RFC3530\]](#)), and NCP (for more information, see [\[NOVELL\]](#)).

2.2.1.3 AD DS Domain Name

Unless specified otherwise, an AD DS Domain Name MUST be a null-terminated Unicode character string consisting of the name of an Active Directory domain services (AD DS) domain in an AD DS forest. This can be either a NetBIOS name or a fully qualified name, as specified in [\[MS-ADTS\]](#).

2.2.1.4 UNC Path

A UNC path can be used to access network resources, and MUST be a null-terminated Unicode character string whose format MUST be \\<hostname>\<sharename>[\<objectname>]*, where <hostname> is the host name of a server or the [AD DS domain name](#) of an Active Directory domain services (DS) domain that hosts resources, or an IP address; <sharename> is the share name or the name of the resource being accessed; and <objectname> is the name of an object, and is dependent on the actual resource being accessed. The notation "[\<objectname>]*" indicates that there may be zero or more object names in the path and each <objectname> is separated from the <objectname> immediately preceding it with the path separator \ (backslash). In a UNC path used to access files and directories in an SMB share, for example, <objectname> may be the name of a file or a directory.

<hostname>, <sharename> and <objectname> are referred to as "pathname components" or "path components". A valid UNC path MUST contain two or more path components. <hostname> is referred to as the "first pathname component", <sharename> as the "second pathname component", and so on. The last component of the path is also referred to as the "leaf component".

The size and valid characters for a path component are defined by the protocol used to access the resource and the type of resource being accessed. The only limitation placed on path components by **DFS** is that path components MUST be at least one character in length and MUST NOT contain a backslash (\) or a null.

2.2.1.5 DFS Root

A DFS root MUST be in one of the following UNC path formats.

- \\<ServerName>\<DFSName>
- \\<DomainName>\<DFSName>

<ServerName> is the host name (as specified in section [2.2.1.1](#)) of a DFS root target (as specified in section [2.2.1.7](#)) of the DFS namespace; <DomainName> is the [AD DS domain name \(section 2.2.1.3\)](#) of the AD DS domain that hosts the domain-based DFS namespace; and <DFSName> is the **DFS namespace name**. Stand-alone DFS namespaces MUST be referred to only by the first path format; domain-based DFS namespaces MUST be referred to in either format, although the second format is the preferred one.

2.2.1.6 DFS Link

A DFS link MUST be in one of the following UNC path formats.

- \\<ServerName>\<DFSName>\<LinkPath>
- \\<DomainName>\<DFSName>\<LinkPath>

<ServerName> is the host name of a DFS root target of the DFS namespace; <DomainName> is the [AD DS domain name](#) of the AD DS domain that hosts the domain-based DFS namespace; <DFSName> is the DFS namespace name, <LinkPath> is the path of the DFS link relative to the DFS root target share. For example, if the full path used to access a DFS-based resource is \\contoso\share1\x\y\mydata.txt, the DFS root target share is \\contoso\share1 and the link path is x\y\mydata.txt. Stand-alone DFS namespaces MUST be referred to only by the first format; domain-based DFS namespaces MUST be referred to in either format, although the second format is the preferred one.

2.2.1.7 DFS Root Target

A DFS root target is a UNC path that MUST be in the following format.

- \\<servername>\<sharename>

<servername> is the hostname of a DFS root target server and <sharename> is the sharename corresponding to a DFS namespace on the DFS root target server.

2.2.1.8 DFS Link Target

A DFS link target is any UNC path that resolves to a directory.

2.2.1.9 DFS Target

A DFS target is either a DFS root target or a DFS link target.

2.2.2 SMB2_REQ_GET_DFS_REFERRAL

A DFS client sends a DFS referral request using the SMB2_REQ_GET_DFS_REFERRAL message. The format of this message is as follows:

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
MaxReferralLevel																RequestFileName (variable)															
...																															

MaxReferralLevel (2 bytes): A 16-bit integer that indicates the highest DFS referral version understood by the client. The DFS referral versions specified by this document are 1 through 4 inclusive. A DFS client **MUST** support DFS referral version 1 through the version number set in this field. The referral response messages are referral version dependent, and are specified in sections [2.2.4.1](#) through [2.2.4.4](#). Depending on the highest DFS referral version understood by the server, the server **MAY** return a response with a referral version less than **MaxReferralLevel**. The server **MUST NOT** return a response containing a referral number greater than **MaxReferralLevel**.

RequestFileName (variable): A null-terminated Unicode string specifying the path to be resolved. The specified path **MUST NOT** be case-sensitive. Its format depends on the type of referral request:

Domain Referral: The path **MUST** be an empty string (containing just the null terminator). A client **MUST** use DFS referral version 3 or later for a domain referral request.

DC Referral: The path **MUST** be "[\<domain>](#)", where [<domain>](#) is an [AD DS domain name](#) which **MUST** be in either NetBIOS or fully qualified domain name forms. The format of the response path **MUST** match the format of the request path. For example, if the request path is in NetBIOS form, the response path **MUST** also be in NetBIOS form. A client **MUST** use DFS referral version 3 or later for a DC referral request.

Sysvol Referral: The path **MUST** be either "[\<domain>\SYSVOL](#)" or "[\<domain>\NETLOGON](#)", where [<domain>](#) is an AD DS domain name which **MUST** be in either NetBIOS or fully qualified domain name forms. The format of the response path **MUST** match the format of the request path. For example, if the request path is in NetBIOS form, the response path **MUST** also be in NetBIOS form. A client **MUST** use DFS referral version 3 or later for a sysvol referral request.

Root Referral: The path **MUST** be either of the form "[\<domain>\<dfsname>](#)" or the form "[\<server>\<dfsname>](#)", where [<domain>](#) is the name of the AD DS domain that hosts the DFS namespace, [<dfsname>](#) is the name of a DFS namespace, and [<server>](#) is a DFS root target host name. NetBIOS and fully qualified domain names **MUST** be supported.

Link Referral: The path **MUST** be either of the form "[\<domain>\<dfsname>\<linkpath>](#)" or the form "[\<server>\<dfsname>\<linkpath>](#)", where [<domain>](#) is the name of the AD DS domain that hosts the DFS namespace, [<dfsname>](#) is the name of a DFS namespace, [<server>](#) is a DFS root target host name, [<linkpath>](#) is a path which may have a DFS link. NetBIOS and fully qualified domain names **MUST** be supported.

2.2.3 RESP_GET_DFS_REFERRAL

A DFS server responds to a DFS client referral request with the RESP_GET_DFS_REFERRAL message. This is referred to as the "referral header" in this document. The format of this message is as follows:

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
PathConsumed																NumberOfReferrals															
ReferralHeaderFlags																															
ReferralEntries (variable)																															
...																															
Padding (variable)																															
...																															

PathConsumed (2 bytes): A 16-bit integer indicating the number of bytes—not characters—in the prefix of the referral request path corresponding to a DFS root or DFS link. This field **MUST** consist of the length of the whole path components in their entirety. For example, if the referral request path is "\\MyDomain\\MyDfs\\dir1\\link1\\dir2\\file1" and "\\MyDomain\\MyDfs\\dir\\link1" is a DFS link, **PathConsumed** is 50 (2 bytes for each of the 25 characters).

NumberOfReferrals (2 bytes): A 16-bit integer indicating the number of referral entries immediately following the referral header.

ReferralHeaderFlags (4 bytes): A 32-bit field representing a series of flags that are combined by using the bitwise OR operation. Only the R, S, and T bits are defined and used. The other bits **MUST** be set to 0 by the server and ignored upon receipt by the client.

Value	Meaning
R 0x04000000	R (ReferralServers): The R bit MUST be set to 1 if all of the targets in the referral entries returned are DFS root targets capable of handling DFS referral requests and set to 0 otherwise.
S 0x02000000	S (StorageServers): The S bit MUST be set to 1 if all of the targets in the referral response can be accessed without requiring further referral requests and set to 0 otherwise.
T 0x01000000	T (TargetFailback): The T bit MUST be set to 1 if DFS client target failback is enabled for all targets in this referral response. This value MUST be set to 0 by the server and ignored by the client for all DFS referral versions except DFS referral version 4.

ReferralEntries (variable): As many [DFS_REFERRAL_V1](#), [DFS_REFERRAL_V2](#), [DFS_REFERRAL_V3](#), or [DFS_REFERRAL_V4](#) structures as indicated by the **NumberOfReferrals** field.

Padding (variable): After DFS_REFERRAL_V3 entries, the server MAY insert padding bytes that MUST be ignored by the client. [<2>](#)

2.2.4 Referral Entry Types

The DFS: Referral Protocol defines four structures used to encode referral entries: [DFS_REFERRAL_V1 \(section 2.2.4.1\)](#), [DFS_REFERRAL_V2 \(section 2.2.4.2\)](#), [DFS_REFERRAL_V3 \(section 2.2.4.3\)](#), and [DFS_REFERRAL_V4 \(section 2.2.4.4\)](#).

All referral entries in a [RESP_GET_DFS_REFERRAL](#) message MUST use the same referral entry structure. As a consequence, all referral entries in a RESP_GET_DFS_REFERRAL message MUST have the same version number.

Each referral entry structure has a 16-bit **Size** field. The **Size** field indicates the total size, in bytes, of the referral entry. Clients MUST add value **Size** field to the offset of a referral entry to find the offset of the next referral entry in the RESP_GET_DFS_REFERRAL message.

The DFS_REFERRAL_V2, DFS_REFERRAL_V3, and DFS_REFERRAL_V4 structures contain fields with offsets to strings. Clients MUST add the string offset to the offset of the beginning of the referral entry to find the offset of the string in the RESP_GET_DFS_REFERRAL message. The strings referenced from the fields of a referral entry MAY immediately follow the referral entry structure or MAY follow the last referral entry in the RESP_GET_DFS_REFERRAL message. The **Size** field of a referral entry structure MUST include the size in bytes of all immediately following strings so that a client can find the next referral entry in the message. The **Size** field of a referral entry structure MUST NOT include the size of referenced strings located after the last referral entry in the message. [<3>](#)

Examples of referral response packets of DFS referral versions 2 and 3 are specified in [section 4](#).

2.2.4.1 DFS_REFERRAL_V1

The format of the version 1 referral entry is as follows:

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
VersionNumber																Size															
ServerType																ReferralEntryFlags															
ShareName (variable)																															
...																															

VersionNumber (2 bytes): A 16-bit integer indicating the version number of the referral entry. MUST always be 0x0001 for DFS_REFERRAL_V1.

Size (2 bytes): A 16-bit integer indicating the total size of the referral entry in bytes.

ServerType (2 bytes): A 16-bit integer indicating the type of server hosting the target. The server returning the referral response MUST set this field to 0x0001 if root targets are returned or to 0x0000 if link targets are returned. It MUST be set to 0x0000 for a sysvol referral response also.

Value	Meaning
0x0001	Root targets returned.
0x0000	Link targets returned or a sysvol referral response.

ReferralEntryFlags (2 bytes): A series of bit flags. MUST be set to 0x0000 and ignored on receipt.

ShareName (variable): A null-terminated Unicode character string that specifies a DFS target.

2.2.4.2 DFS_REFERRAL_V2

The format of the version 2 referral entry is as follows:

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
VersionNumber																Size															
ServerType																ReferralEntryFlags															
Proximity																															
TimeToLive																															
DFSPathOffset																DFSAlternatePathOffset															
NetworkAddressOffset																															

VersionNumber (2 bytes): A 16-bit integer indicating the version number of the referral entry. MUST always be 0x0002 for DFS_REFERRAL_V2.

Size (2 bytes): A 16-bit integer indicating the total size of the referral entry in bytes.

ServerType (2 bytes): A 16-bit integer indicating the type of server that hosts the target. The server that returns the referral response MUST set this field to 0x0001 if root targets are returned or to 0x0000 if link targets are returned. It MUST be set to 0x0000 for sysvol referral response also.

Value	Meaning
0x0001	Root targets returned.
0x0000	Link targets returned or a sysvol referral response.

ReferralEntryFlags (2 bytes): MUST be set to 0x0000 by the server and ignored on receipt by the client.

Proximity (4 bytes): MUST be set to 0x00000000 by the server and ignored by the client.

TimeToLive (4 bytes): A 32-bit integer indicating the time-out value, in seconds, of the DFS root or DFS link. MUST be set to the time-out value of the DFS root or the DFS link in the DFS metadata for which the referral response is being sent. When there is more than one referral entry, the **TimeToLive** of each referral entry MUST be the same.

DFSPathOffset (2 bytes): A 16-bit integer indicating the offset, in bytes, from the beginning of this referral entry to the DFS path that corresponds to the DFS root or the DFS link for which target information is returned.

DFSAlternatePathOffset (2 bytes): A 16-bit integer indicating the offset, in bytes, from the beginning of this referral entry to the DFS path that corresponds to the DFS root or the DFS link for which target information is returned. This path MAY either be the same as the path as pointed to by the **DFSPathOffset** field or be an **8.3 name**. In the former case, the string referenced MAY be the same as that in the **DFSPathOffset** field or a duplicate copy. [<4>](#)

NetworkAddressOffset (2 bytes): A 16-bit integer indicating the offset, in bytes, from beginning of this referral entry to the DFS target path that correspond to this entry.

2.2.4.3 DFS_REFERRAL_V3

2.2.4.3.1 DFS_REFERRAL_V3 (NameListReferral Bit Set to 0)

When the NameListReferral bit in the **ReferralEntryFlags** field is 0, the format of the version 3 referral entry MUST be as follows:

0	1	2	3	4	5	6	7	8	9	10	1	2	3	4	5	6	7	8	9	20	1	2	3	4	5	6	7	8	9	30	1
VersionNumber																Size															
ServerType																ReferralEntryFlags															
TimeToLive																															
DFSPathOffset																DFSAlternatePathOffset															
NetworkAddressOffset																ServiceSiteGuid															
...																															
...																															
...																															
...																															

VersionNumber (2 bytes): A 16-bit integer indicating the version number of the referral entry. MUST always be 0x0003 for [DFS_REFERRAL_V3](#).

Size (2 bytes): A 16-bit integer indicating the total size of the referral entry, in bytes.

ServerType (2 bytes): A 16-bit integer indicating the type of server that hosts the target. The server that returns the referral response MUST set this field to 0x0001 if root targets are returned in this entry or to 0x0000 if link targets are returned. It MUST be set to 0x0000 for a sysvol referral response also.

Value	Meaning
0x0001	Root targets returned.
0x0000	Link targets returned or a sysvol referral response.

ReferralEntryFlags (2 bytes): A 16-bit field representing a series of flags that are combined by using the bitwise OR operation. Only the N bit is defined for DFS_REFERRAL_V3. The other bits MUST be set to 0 by the server and ignored upon receipt by the client.

Value	Meaning
N 0x0200	N (NameListReferral): MUST be set for a domain referral response or a DC referral response.

TimeToLive (4 bytes): A 32-bit integer indicating the time-out value, in seconds, of the DFS root or DFS link. MUST be set to the time-out value of the DFS root or DFS link in the DFS

metadata for which the referral response is being sent. When there is more than one referral entry, the **TimeToLive** field of each referral entry **MUST** be the same.

DFSPathOffset (2 bytes): A 16-bit integer indicating the offset, in bytes, from the beginning of this referral entry to the DFS path that corresponds to the DFS root or DFS link for which target information is returned. The DFS path **MUST** be a null-terminated string.

DFSAlternatePathOffset (2 bytes): A 16-bit integer indicating the offset, in bytes, from the beginning of this referral entry to the DFS path that corresponds to the DFS root or DFS link for which target information is returned. This path **MAY** either be the same as the path as pointed to by the **DFSPathOffset** field or be an 8.3 name. In the former case, the string referenced **MAY** be the same as that in **DFSPathOffset** or a duplicate copy.<5>

NetworkAddressOffset (2 bytes): A 16-bit integer indicating the offset, in bytes, from the beginning of this referral entry to the DFS target path that corresponds to this entry. The DFS path **MUST** be a null-terminated string.

ServiceSiteGuid (16 bytes): These 16 bytes **MUST** always be set to 0 by the server and ignored by the client. For historical reasons, this field was defined in early implementations, but never used.

2.2.4.3.2 DFS_REFERRAL_V3 (NameListReferral Bit Set to 1)

When the NameListReferral bit in the **ReferralEntryFlags** field is 1, the format of the version 3 referral entry is as follows:

Server and client **MUST** use the following fields when the NameListReferral bit is 1.

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
VersionNumber																Size															
ServerType																ReferralEntryFlags															
TimeToLive																															
SpecialNameOffset																NumberOfExpandedNames															
ExpandedNameOffset																															

VersionNumber (2 bytes): A 16-bit integer indicating the version number of the referral entry. **MUST** always be 0x0003 for [DFS_REFERRAL_V3](#).

Size (2 bytes): A 16-bit integer indicating the total size of the referral entry, in bytes.

ServerType (2 bytes): A 16-bit integer indicating the type of server that hosts the target. The server that returns the referral response **MUST** set this field to 0x0001 if root targets are returned in this entry or to 0x0000 if link targets are returned. It **MUST** be set to 0x0000 for a sysvol referral response also.

Name	Value
root targets returned	0x0001
link targets returned or sysvol referral response	0x0000

ReferralEntryFlags (2 bytes): Represents a series of flags. Only the N bit is defined for DFS_REFERRAL_V3. The other bits MUST be set to 0 by the server, and ignored upon receipt by the client.

Value	Meaning
N 0x0200	N (NameListReferral): MUST be set for a domain referral response or a DC referral response.

TimeToLive (4 bytes): A 32-bit integer indicating the time-out value, in seconds, of the DFS root or DFS link. MUST be set to the time-out value of the DFS root or DFS link in the DFS metadata for which the referral response is being sent. When there is more than one referral entry, the **TimeToLive** field of each referral entry MUST be the same.

SpecialNameOffset (2 bytes): A 16-bit integer indicating the offset, in bytes, from the beginning of the referral entry to a domain name. For a domain referral response, this MUST be the [AD DS domain name](#) that corresponds to the referral entry. For a DC referral response, this MUST be the AD DS domain name that is specified in the DC referral request. The domain name MUST be a null-terminated string.

NumberOfExpandedNames (2 bytes): A 16-bit integer indicating the number of DCs being returned for a DC referral request. MUST be set to 0 for a domain referral response.

ExpandedNameOffset (2 bytes): A 16-bit integer indicating the offset, in bytes, from the beginning of this referral entry to the first DC name string returned due to a DC referral request. For a DC referral, the first DC name string must be followed immediately by the additional DC name strings so that there are a total of the **NumberOfExpandedNames** field's consecutive strings. This field MUST be set to 0 for a domain referral response. The DC name MUST be a null-terminated string.

2.2.4.4 DFS_REFERRAL_V4

The format of the version 4 referral entry MUST be exactly the same as the format of the [DFS_REFERRAL_V3](#) referral entry, with two exceptions.

The value of the **VersionNumber** field MUST be 0x0004 for DFS_REFERRAL_V4.

The **ReferralEntryFlags** field contains an additional flag that can be set as follows:

Value	Meaning
T 0x0400	T (TargetSetBoundary): MUST be set if the target corresponding to this referral entry is the first target of a target set . The first referral entry MUST have this bit set. A target set is defined as a consecutive list of referral entries, where the first entry has the T bit set, and the last entry is either the last entry in the DFS referral response, or it is immediately followed by a DFS referral entry with the T bit set.

The other bits of the **ReferralEntryFlags** field MUST be set to 0 by the server and ignored upon receipt by the client.

3 Protocol Details

The following sections specify details of the Distributed File System (DFS): Referral Protocol, including abstract data models, higher-layer triggered events, and message processing events and sequencing rules.

3.1 DFS Client Details

In this section, unless stated otherwise, the term "client" refers to a DFS client. [<6>](#)

The following flow charts show the steps taken by a DFS client to resolve a DFS path.

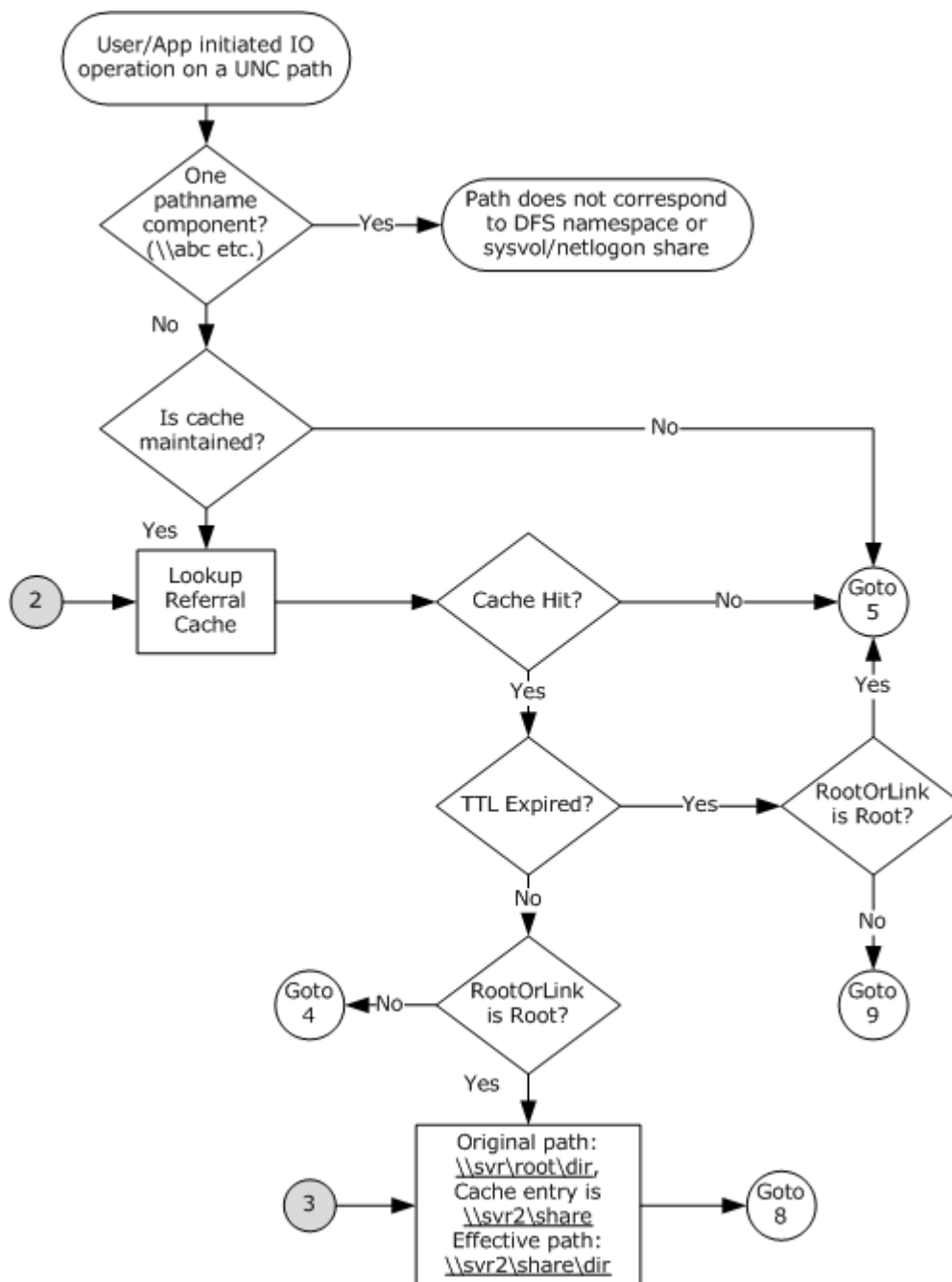


Figure 1: DFS Path resolution - initial steps

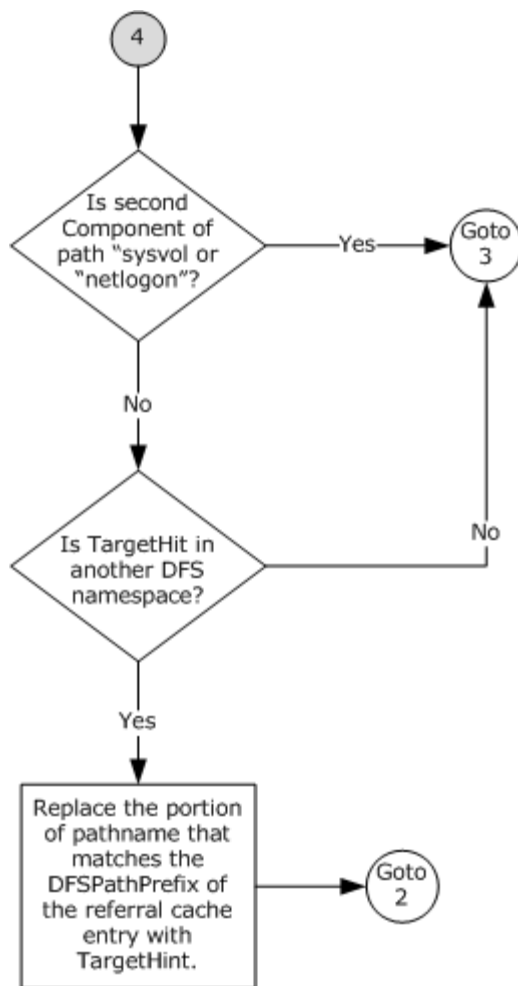


Figure 2: DFS Path resolution - connection 4

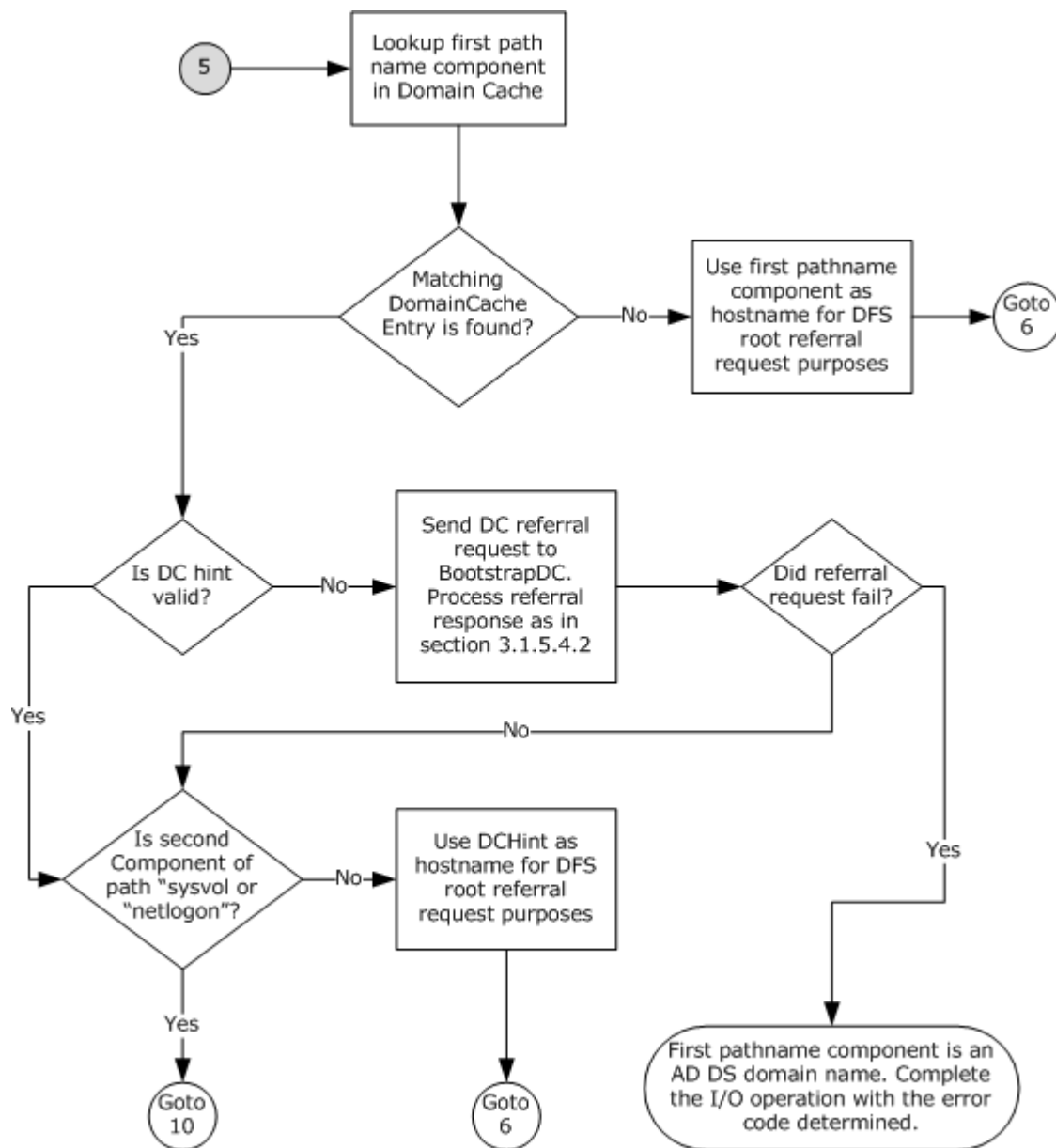


Figure 3: DFS Path resolution - connection 5

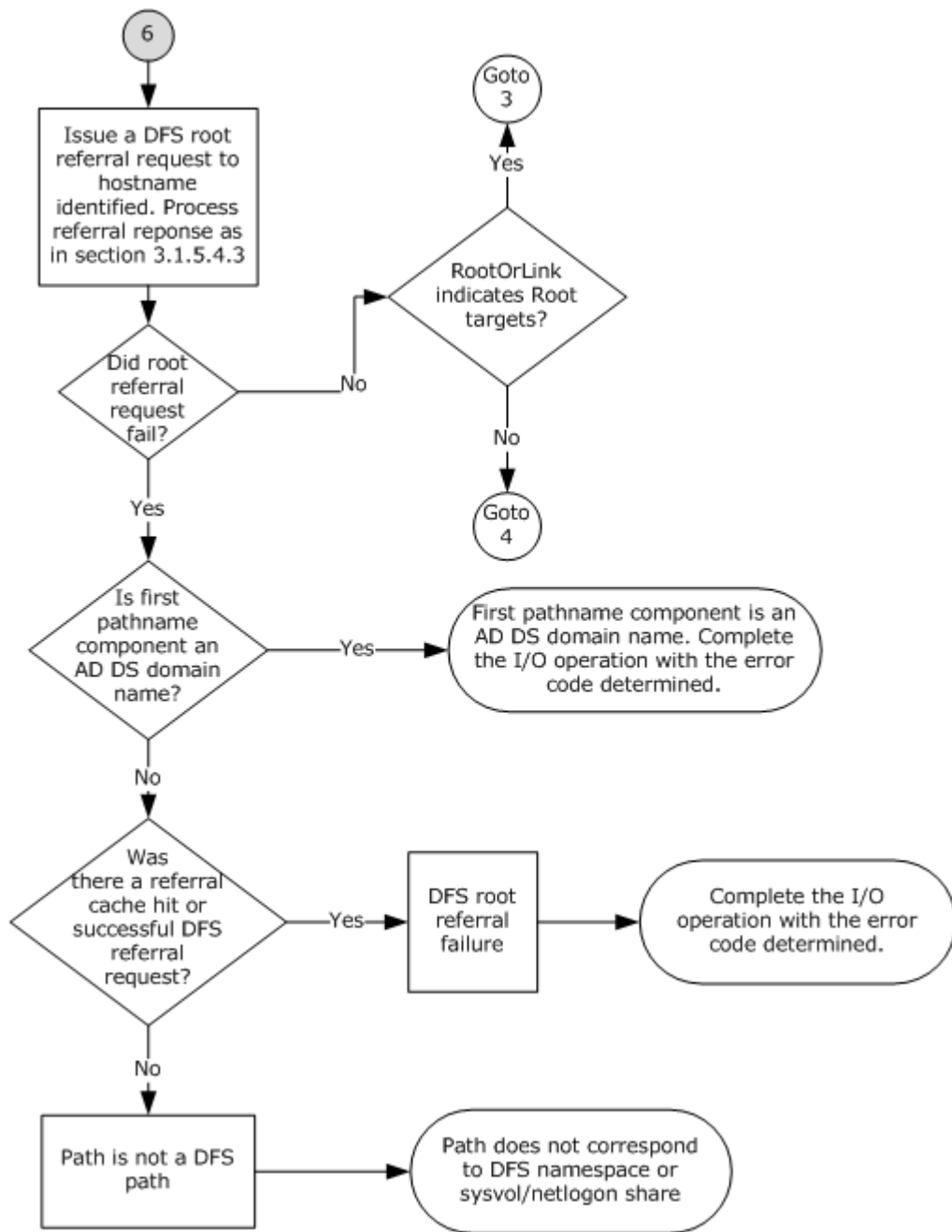


Figure 4: DFS Path resolution - connection 6

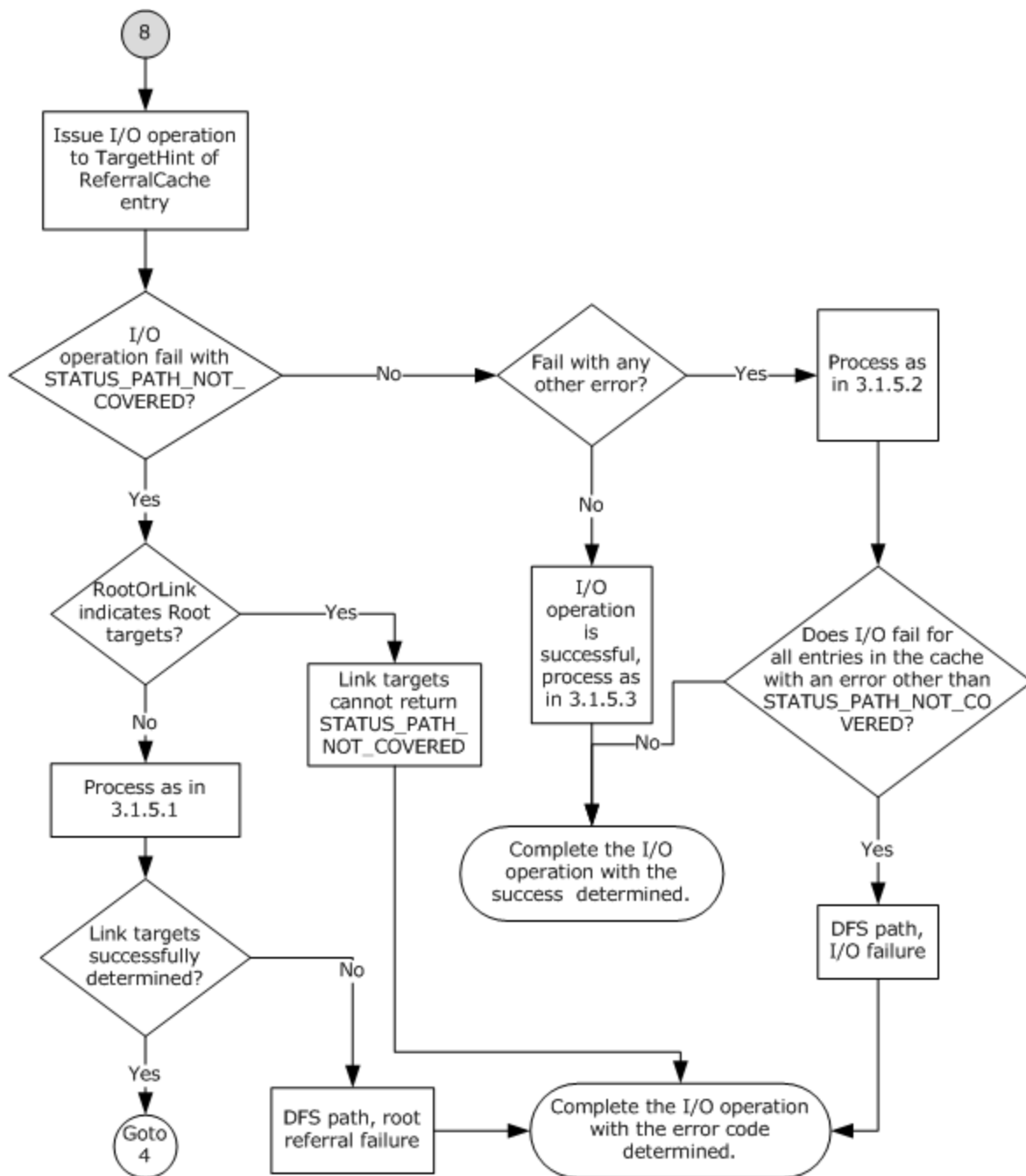


Figure 5: DFS Path resolution - connection 8

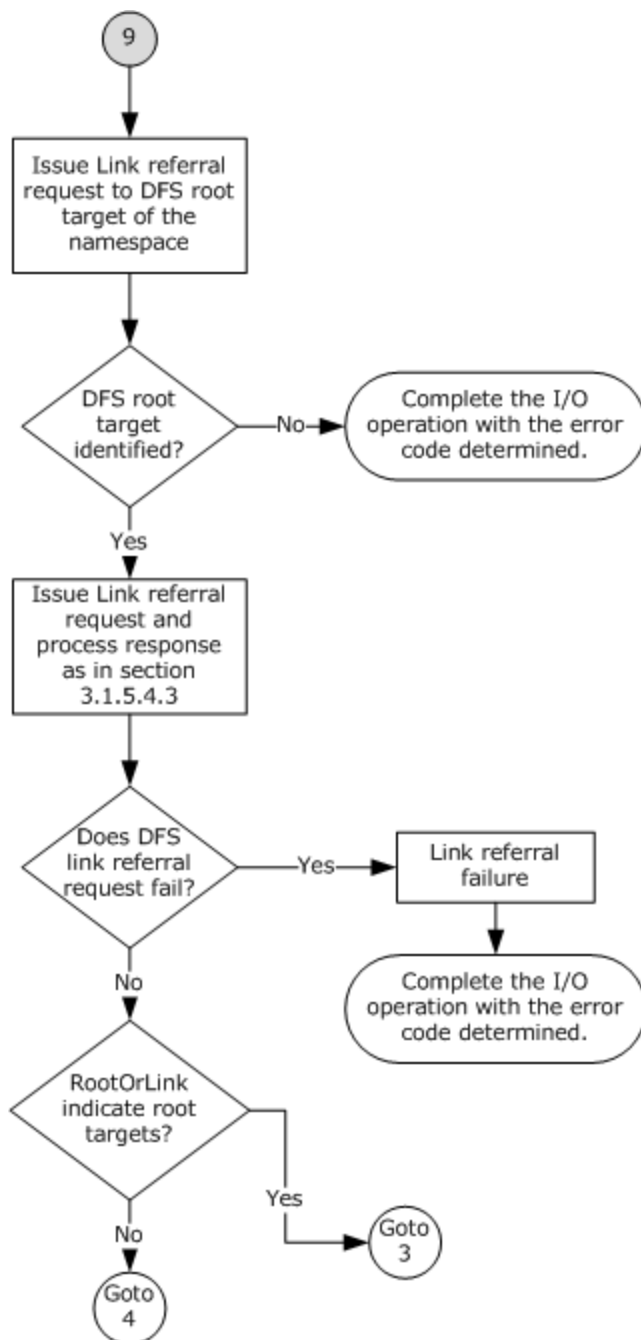


Figure 6: DFS Path resolution - connection 9

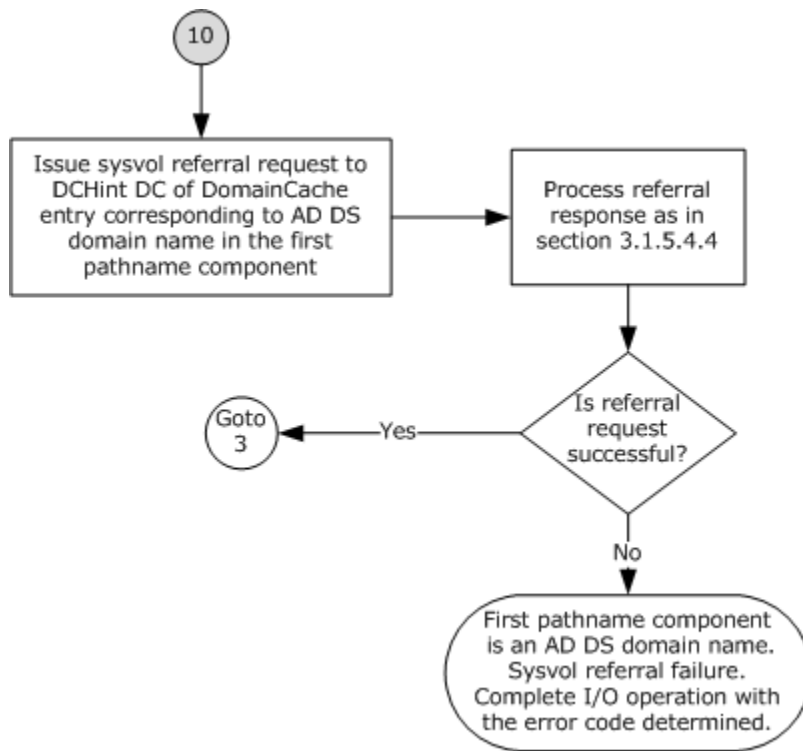


Figure 7: DFS Path resolution - connection 10

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 organization is provided to explain how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document.

Because the DFS clients effectively block an application from accessing a specific file share until the DFS: Referral Protocol can map the DFS root name to an actual file server path, it is advantageous to cache DFS referral responses. Because of this, clients MAY maintain local caches of information received through referral requests to avoid future referral requests and to improve the performance of DFS resource access. [<7>](#)

- BootstrapDC: Applicable only for a computer joined to a domain. This is a DC from which the DFS client MUST be able to obtain a list of trusted AD DS domains, in addition to a list of DC hostnames for an AD DS domain.
- DomainCache: Applicable only for a computer joined to an AD DS domain. MUST contain a list of trusted AD DS domains in both NetBIOS and fully qualified domain name forms, in addition to a list of DC hostnames for each AD DS domain. Conceptually, this is an array of tuples of the form <ADDSDomainName, DCHint, DCList>. Cache lookup involves finding a DomainCache entry with a matching ADDSDomainName. This can be used to check for a valid AD DS domain name or to find a DC hostname for a given AD DS domain name. DCHint identifies a DC hostname from DCList which MUST be the DC that was last successfully used by the DFS client.
- ReferralCache: MUST contain a cache of root, link, and sysvol referral responses. A hit on a referral cache entry indicates that the path in a name resolution operation is a DFS namespace or

a SYSVOL/NETLOGON share. On a miss, the DFS client MUST determine whether the path in a name resolution operation is a DFS path or a SYSVOL/NETLOGON share, as described below.

A referral cache entry conceptually contains entries indexed by a DFS path prefix, DFSPathPrefix. An entry is a tuple of the form <DFSPathPrefix, RootOrLink, TTL, TargetFailback, TargetHint, TargetList>.

DFSPathPrefix MUST be a DFS root path or a DFS Link path, and MUST be the same as the string pointed to by the **DFSPathOffset** of a [DFS_REFERRAL_V2](#), [DFS_REFERRAL_V3](#) or [DFS_REFERRAL_V4](#) referral entry.

RootOrLink MUST identify whether the entry contains DFS root targets or DFS link targets. It MUST contain the value from the **ServerType** field of a referral entry (as specified in sections [2.2.4.1](#), [2.2.4.2](#), [2.2.4.3](#), and [2.2.4.4](#)).

TargetFailback is used only for DFS_REFERRAL_V4 and contains the value from the TargetFailback bit of the referral header (as specified in section [2.2.3](#)).

TTL contains a value derived from the **TimeToLive** field of a referral entry (as specified in sections [2.2.4.1](#), [2.2.4.2](#), [2.2.4.3](#), and [2.2.4.4](#)). This MUST be a time stamp at which a referral cache entry is considered to be expired. An implementation is free to come up with soft and hard time-outs based on the **TimeToLive** field of the referral entry, for example. The soft time-out may be used to initiate a referral cache entry refresh operation while permitting the use of the referral cache entry; the hard time limit may be used to fail any operation using the referral cache entry should all attempts to refresh it fail. [<8>](#)

TargetHint MUST identify a target in TargetList which was last successfully used by the DFS client. TargetList consists of tuples of the form <TargetPath, TargetSetBoundary>, where TargetPath MUST be the string pointed to by the **NetworkAddressOffset** (as specified in sections [2.2.4.2](#), [2.2.4.3](#), and [2.2.4.4](#)). TargetSetBoundary MUST be used only for V4 referrals and MUST contain the value from the TargetSetBoundary of the referral entry (as specified in section [2.2.4.4](#)).

A lookup in ReferralCache MUST involve searching for an entry with DFSPathPrefix which is a complete prefix of the path being looked up. Whole path components MUST match. Also, the ReferralCache entry with the longest match MUST be used. For the lookup path of \MyDomain\MyDfs\MyDir\file1, if there are two ReferralCache entries, one for \MyDomain\MyDfs and another for \MyDomain\MyDfs\MyDir, the latter is the longest match.

3.1.2 Timers

BootstrapDCTimer: This timer is applicable only to a domain-joined client. This timer is used to regularly update the BootstrapDC. [<9>](#)

3.1.3 Initialization

A domain-joined client MUST initialize BootstrapDC and BootstrapDCTimer.

3.1.4 Higher-Layer Triggered Events

3.1.4.1 User/Application Initiated I/O Operation on a UNC Path

When the DFS client receives an I/O operation with a UNC path, it MUST perform a name resolution operation to determine whether a DFS namespace or the SYSVOL/NETLOGON share is being accessed, as detailed in the following sequence of operations. Refer to the flow chart in section

[3.1](#) for a graphical illustration of the process. In some cases, individual steps have a description of the condition at entry of the step inside square brackets [like this].

The following procedure specifies a possible implementation for resolving a UNC path. This is how Windows resolves the UNC path. This scenario uses a cache on the DFSC client to avoid future referral requests, thereby improving the performance of the DFS client. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document. If a client implementation chooses to not implement a client side cache, it MUST request information from the server and use the referral response entry to resolve the names.

The sequence of operations used by Windows MUST be as follows:

1. If the path has only one path component (for example, \\abc), go to step 12.
2. Look up the path in ReferralCache if a cache is being maintained. If no cache is being maintained, go to step 5.
 1. If no matching entry is found in ReferralCache, go to step 5.
 2. If an entry's TTL has expired:
 - Go to step 5 if RootOrLink indicates DFS root targets.
 - RootOrLink indicates DFS link targets. Go to step 9.
 3. If an entry contains DFS link targets as indicated by RootOrLink, go to step 4.
3. [ReferralCache hit, unexpired TTL, RootOrLink=root] Replace the portion of the path that matches DFSPathPrefix of ReferralCache entry with the DFS target path of TargetHint of ReferralCache entry. For example, if the path is \\MyDomain\\MyDfs\\MyDir and ReferralCache entry contains \\MyDomain\\MyDfs with a DFS target path of \\someserver\\someshare\\somepath, the effective path becomes \\someserver\\someshare\\somepath\\MyDir. Go to step 8.
4. [ReferralCache hit, unexpired TTL, RootOrLink=link]
 1. If second component of path is "sysvol" or "netlogon" go to step 3.
 2. Check if TargetHint is in another DFS namespace by using its first two path components, and doing a lookup in ReferralCache or issuing DFS root referral request as required:
 - If RootOrLink of ReferralCache entry corresponding to the first two components of TargetHint indicates root targets, TargetHint is in another DFS namespace, go to step 11.
 - If TargetHint is not in another DFS namespace, go to step 3.
5. [ReferralCache miss] [ReferralCache hit, expired TTL, RootOrLink=root] Lookup first path component in DomainCache.
 1. If no matching DomainCache entry is found, use first path component as hostname for DFS root referral request purposes. Go to step 6.
 2. If DCHint is not valid, send DC referral request to BootstrapDC for ADDSDomainName. The process referral response is as specified in section [3.1.5.4.2](#). If the referral request fails, go to step 13.
 3. If second path component is sysvol or netlogon, go to step 10.

4. Use DCHint as hostname for DFS root referral request purposes.
6. [DFS Root referral request] Issue DFS root referral request to hostname identified. The process referral response is as specified in section [3.1.5.4.3](#). On DFS root referral request failure:
 1. If the first path component is an AD DS domain name, go to step 13.
 2. If there was a ReferralCache hit or a successful DFS referral request at any time during the processing of this I/O request, path is in a DFS namespace. Go to step 15.
 3. The path is not a DFS path and no further processing is required. Go to step 12.
7. [DFS root referral success] If ReferralCache entry's RootOrLink indicates root targets, go to step 3; otherwise, go to step 4.
8. [I/O request, path fully resolved] Issue I/O operation to TargetHint of ReferralCache entry.
 1. If I/O operation fails with STATUS_PATH_NOT_COVERED.
 - RootOrLink of ReferralCache entry indicates link targets, go to step 14.
 - If RootOrLink of ReferralCache entry indicates root targets, the process is as specified in section [3.1.5.1](#). If link target cannot be successfully determined, go to step 15.
 - Link target determined successfully. Go to step 4.
 2. If I/O operation fails with an error other than STATUS_PATH_NOT_COVERED, process is as specified in section [3.1.5.2](#). If I/O to all targets in TargetList of ReferralCache entry failed with an error other than STATUS_PATH_NOT_COVERED, go to step 16.
 3. I/O operation is successful; process is as specified in section [3.1.5.3](#). Go to step 17 and complete I/O operation with success.
9. [ReferralCache hit, expired TTL, RootOrLink=link] Link referral request must be issued to a DFS root target of the namespace. Find ReferralCache entry corresponding to the first two path components, issuing a DFS root referral request if required. If no DFS root target can be identified, it is a root referral failure; go to step 17 and fail I/O request. Otherwise, issue DFS Link referral request to DFS root target and process DFS referral response as specified in section [3.1.5.4.3](#). If DFS Link referral request fails, go to step 15. Otherwise:
 1. If RootOrLink of refreshed ReferralCache entry indicates DFS root targets, go to step 3.
 2. RootOrLink of refreshed ReferralCache entry indicates DFS link targets. Go to step 4.
- 10.[sysvol referral request] Issue sysvol referral request to DCHint DC of DomainCache entry that corresponds to the AD DS domain name in the first path component. Process referral response is as specified in section [3.1.5.4.4](#). If the referral request is successful, go to step 3; otherwise, go to step 13.
- 11.[interlink] Replace the portion of the path that matches the DFSPathPrefix of the ReferralCache entry with TargetHint. For example, if the path is \MyDomain\MyDfs\MyLink\MyDir and the referral entry contains \MyDomain\MyDfs\MyLink with a DFS target path of \someserver\someshare\somepath, the effective path becomes \someserver\someshare\somepath\MyDir. Go to step 2.
- 12.[not DFS] Path does not correspond to a DFS namespace or a sysvol/netlogon share, no further processing is required on the user/application initiated I/O request by the DFS client.

- 13.[Cannot get DC for domain][sysvol referral failure] First path component is an AD DS domain name. Go to step 17 and fail the I/O operation with the last error code that occurred before the jump to this step.
- 14.[STATUS_PATH_NOT_COVERED from link target] Link targets cannot return STATUS_PATH_NOT_COVERED. Go to step 17 and fail the I/O operation with the last error code that occurred before the jump to this step.
- 15.[Link referral failure] [DFS path, root referral failure] Path is known to be in a DFS namespace, but DFS root referral request or DFS Link referral request has failed. Go to step 17 and fail the I/O operation with the last error code that occurred before the jump to this step. [Windows<10>](#)
- 16.[DFS path, I/O failure] Path is known to be in a DFS namespace but I/O operation to all the targets in TargetList of ReferralCache entry failed. Go to step 17 and fail the I/O operation with the last error code that occurred before the jump to this step. [.<11>](#)
- 17.[complete I/O request] Complete the user/application initiated I/O request with the success or error code that occurred before the jump to this step. [.<12>](#)

The CAP_DFS flag in the SMB_COM_NEGOTIATE Server Response extension (as specified in section [2.1](#)) is set when the server is DFS aware. A DFS client MAY use this information to decide that a path, having as its first path component a server that does not set this flag, is not in a DFS namespace without sending a DFS referral request. [.<13>](#)

For the versions of SMB protocol specified in [\[MS-SMB\]](#), the DFS client MUST send DFS referral requests in the TRANS2_GET_DFS_REFERRAL subcommand of the SMB_COM_TRANSACTION2 SMB command and wait synchronously for its completion (as specified in [\[MS-SMB\]](#) section 3.2.4.12). For the versions of SMB protocol specified by [\[MS-SMB2\]](#), the DFS client MUST send DFS referral requests (as specified in [\[MS-SMB2\]](#) section 3.2.4.19.3) and MUST wait synchronously for its completion.

3.1.5 Message Processing Events and Sequencing Rules

3.1.5.1 I/O Operation to Target Fails with STATUS_PATH_NOT_COVERED

When an I/O operation issued to a DFS root target server in Step 8 of section [3.1.4.1](#) fails with STATUS_PATH_NOT_COVERED (0xC0000257), it indicates that the portion of the DFS namespace accessed by the client is not contained in the DFS root target server. When an I/O operation issued to a link target fails with STATUS_PATH_NOT_COVERED (0xC0000257), the client MUST fail the original I/O request.

To identify the DFS link targets that contain the required portion of the DFS namespace, the client MUST look up the path used for the I/O operation in ReferralCache. On a cache miss, it SHOULD issue a DFS link referral request to the DFS root target server specified by TargetHint of ReferralCache entry corresponding to the DFS namespace. If successful, it MUST create a new ReferralCache entry with RootOrLink set to indicate link targets. [.<14>](#)

The client MUST set the **RequestFileName** field of the [SMB2_REQ_GET_DFS_REFERRAL \(section 2.2.2\)](#) structure to the path in the I/O operation issued to the DFS root target in Step 8 of section [3.1.4.1](#). It MUST set the **MaxReferralLevel** field of the SMB2_REQ_GET_DFS_REFERRAL message to the highest DFS referral version supported. [.<15>](#)

3.1.5.2 I/O Operation to Target Fails with an Error Other than STATUS_PATH_NOT_COVERED

When an I/O operation issued to the target server identified by the TargetHint of the ReferralCache entry used fails with specific error codes) as specified in [\[MS-SMB\]](#), the next target from TargetList of ReferralCache entry MUST be selected, if any. If all the targets in ReferralCache entry have been tried, the client MUST fail the user/application initiated I/O operation.

If a target was selected from the TargetList, the I/O operation MUST be issued to the target. This is known as "**DFS client target failover**". The TargetHint of the ReferralCache entry used MUST be set to this target as a performance optimization for another operation using the same ReferralCache entry.

The DFS client MAY support a DFS link that has more than one target in a DFS namespace that is different from the DFS link's namespace.[<16>](#)

3.1.5.3 I/O Operation to a DFS Root Target or DFS Link Target Succeeds

When the operation issued to a DFS root target or DFS link target succeeds, the DFS client MUST complete the user/application initiated I/O operation with the same status code.

3.1.5.4 Receiving a Referral Response

The DFS client receives the referral response in the form of [RESP_GET_DFS_REFERRAL \(section 2.2.3\)](#). If the **NumberOfReferrals** field is at least 1, the client MUST determine the version number of the referral response by accessing the **VersionNumber** field of the first referral entry immediately following the referral header and interpreting the referral entry as specified in section [2.2.4](#).

The **NumberOfReferrals** field in the referral header contains the number of referral entries in the response. The first referral entry immediately follows the referral header [RESP_GET_DFS_REFERRAL](#) (section 2.2.3). From a given referral entry, the next referral entry is accessed by adding the **Size** field of the referral entry to the address of the given referral entry.

If a DFS referral request fails with STATUS_BUFFER_OVERFLOW (0x80000005), the client SHOULD retry with the referral request with a bigger buffer size.[<17>](#)

3.1.5.4.1 Receiving a Domain Referral Response

This is applicable only to a domain-joined computer. The client receives this response for the domain referral request that it issued to BootstrapDC (as specified in section [3.1.6](#)). The domain referral response MUST be version 3 or later; otherwise, the client MUST ignore the referral response.

The client MUST discard referral responses that do not have the NameListReferral bit of each referral entry set. The client can access the null-terminated Unicode [AD DS domain name](#) contained in a referral entry by adding the value in the **SpecialNameOffset** field of the referral entry to the address of the referral entry.

The client MUST add the AD DS domain names received in the referral response to DomainCache, if empty. If DomainCache already has AD DS domain names, the client MUST add only those AD DS domain names to DomainCache that are not already in it. The client MAY remove those AD DS domain names from DomainCache that are not in the referral response.[<18>](#)

The DFS client MUST NOT modify the DomainCache on a domain referral failure.

3.1.5.4.2 Receiving a DC Referral Response

This is applicable only to a domain-joined computer. The DFS client receives this referral response for the DC referral request that it sent in step 5.2 of section [3.1.4.1](#). The DC referral response MUST be version 3 or later; otherwise, the client MUST ignore the referral response.

The client MUST verify that the **NumberOfReferrals** field of the referral header is 1 and that the **NameListReferral** bit is set in the referral entry. The **NumberOfExpandedNames** in the referral entry contains the number of DC names returned. The client can access the first null-terminated Unicode DC name string that is returned by adding the value in the **ExpandedNameOffset** field to the address of the referral entry. Immediately following the null termination of a DC name is the next DC name returned. The client can access the null-terminated Unicode [AD DS domain name](#) that corresponds to the referral response by adding the value in the **SpecialNameOffset** to the address of the referral entry.

The client MUST add the list of DCs determined for an AD DS domain name to DCList of the DomainCache entry that corresponds to the AD DS domain name. If the DomainCache entry's DCList is not empty, the client MUST replace it with the DC list from the referral response and set DCHint to the first DC in the new DCList.

3.1.5.4.3 Receiving a Root Referral Response or Link Referral Response

This section describes the processing that occurs when the client gets a referral response after sending a DFS root referral request in step 6 of section [3.1.4.1](#) or a link referral request in either step 9 of section [3.1.4.1](#) or in section [3.1.5.1](#).

If the referral request is successful, but the **NumberOfReferrals** field in the referral header (as specified in section [2.2.3](#)) is 0, the DFS server could not find suitable targets to return to the client. In this case, the client MUST fail the original I/O operation with an implementation-defined error. [<19>](#)

If the root referral request was sent to a DC (steps 5.4 and 6 of section [3.1.4.1](#)), and it fails with an error, the client MUST issue a referral request to the next DC from the DCList of the DomainCache entry, as in step 6 of section [3.1.4.1](#). If all DCs from the DCList have been tried, the DFS client SHOULD fail the original I/O operation with an implementation-defined error. [<20>](#)

If the domain name is not valid, the DFS client SHOULD fail the original I/O operation with an implementation-defined error. [<21>](#)

When the root referral request sent to a DC is successful, the client MUST set the DCHint of the DomainCache entry used to that DC. The DCHint serves as an optimization and avoids having to determine an available and accessible DC for every operation which uses a given DomainCache entry.

If the client sends a DFS root referral request and receives a referral response with the **ServerType** field of the referral response entry (as specified in section [2.2.4](#)) set to 0x0000, a link referral response is being returned. This can happen when server accessed has been consolidated using the Microsoft File Server Migration Toolkit. For more information, see [\[MSFSMT\]](#). [<22>](#)

The client can access the null-terminated Unicode DFS target path contained in a referral entry by adding the value in the **NetworkAddressOffset** field (as specified in section [2.2.3](#)) of the referral entry to the address of the referral entry. The client can access the null-terminated Unicode DFS root or link path that corresponds to the referral response by adding the value in the **DFSPathOffset** field (as specified in section [2.2.3](#)) of the referral entry to the address of the referral entry.

If this response is due to a referral request that was sent in step 6 of section [3.1.4.1](#) or section [3.1.5.1](#) on a ReferralCache miss, the client MUST create a new ReferralCache entry from the referral response and set TargetHint to the first target in TargetList. For a DFS referral version 4 response, the client MUST set TargetSetBoundary of each target in TargetList to the value of the TargetSetBoundary bit of the referral entry (see section [2.2.4.4](#)). The client MUST set the value of RootOrLink in ReferralCache entry based on the **ServerType** field of the referral entry.

If this response is due to a referral request that was sent in step 2.2 in section [3.1.4.1](#) to refresh an existing but expired ReferralCache entry, the client MUST perform the following steps.

- If the targets contained in the TargetList of the ReferralCache entry and those returned in the referral response are equivalent, the client MUST NOT update the TargetList in the ReferralCache entry. The equivalence check operation is specified as follows:
 - DFS referral version 4 response: The number of target sets is the same, and each target set in the ReferralCache entry and the referral response contain the same targets, independent of their order with the target set.
 - DFS referral version 1, 2, or 3 response: The ReferralCache entry and the referral response contain the same targets, independent of their order.
- Update the TTL and TargetFailback in the ReferralCache entry from the referral response.
- If the target specified by the TargetHint is not present in the referral response, set the TargetHint to point to the first target of the TargetList in ReferralCache.
- If the TargetFailback is set in the ReferralCache entry, and if the TargetHint is not in the first target set of the TargetList, set the TargetHint to point to the first target in the first target set of the TargetList. This is referred to as a DFS client target failback. Target failback may also be performed at instances other than just at ReferralCache entry refresh time based on an implementation-defined policy.

If an attempt to refresh an existing, but expired ReferralCache entry fails, an implementation-defined error behavior MAY be used. For example, the client may initiate ReferralCache entry refresh at the end of a soft time-out period (as specified in section [3.1.1](#)) while permitting the use of the entry and either discard it at the end of a hard time-out period (as specified in section [3.1.1](#)) or fail the I/O operations that use the ReferralCache entry. [<23>](#)

3.1.5.4.4 Receiving a sysvol Referral Response

The client receives this referral response for the sysvol referral request that it sent in step 10 of section [3.1.4.1](#).

This referral response MUST be handled in the same manner as that of a root/link referral response, as specified in section [3.1.5.4.3](#).

3.1.6 Timer Events

BootstrapDCTimer: When this timer fires, the DFS client MUST update BootstrapDC with the name of a DC in the AD DS domain of the client. It MUST then issue a domain referral (as specified in section [2.2.2](#)) to the Bootstrap DC. It MUST then restart the timer.

3.1.7 Other Local Events

On joining or leaving a domain, the client re-evaluates the DFS referral requests applicable to it, as specified in section [1.6](#).

3.2 DFS Root Target Server Details

A DFS root target server hosts the DFS root of a DFS namespace. This section specifies how a DFS root target server processes and responds to DFS referral requests from DFS clients. If the server is also a DC, it MUST also conform to the specification in section [3.3](#).

DFS root target servers respond to DFS root referral requests and DFS link referral requests. [<24>](#)

3.2.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The organization is provided to explain how the protocol behaves. This document does not mandate that implementations adhere to this model as long as their external behavior is consistent with that described in this document. The following state MUST be maintained at the DFS Server:

HomeDomain: Applicable only for a computer joined to a domain. MUST be the name of an AD DS domain to which the computer is joined. Maintained both in NetBIOS and fully qualified domain name forms. This is used to access the DFS metadata of domain-based DFS namespace hosted by the server.

DFSNamespacesList: MUST be a list of domain-based and standalone DFS root targets hosted by the server. Used while processing DFS referral requests from DFS clients.

DFSMetadataCache: Cache that MUST contain the DFS metadata of DFS namespaces for which the server is a root target. The cache MUST be maintained as specified in [\[MS-DFSNM\]](#) section 3.2.

DFS in-site referral mode: A Boolean used to determine whether in-site referrals are enabled for this DFS referral or the entire DFS namespace. If TRUE, in-site referrals are enabled.

3.2.2 Timers

No timers are required by the protocol.

3.2.3 Initialization

When the DFS server is started:

- It MUST notify the Server Message Block (SMB) server for reasons as specified in [\[MS-SMB\]](#) section 3.3.4.3. The exact means of how this is accomplished is outside the scope of this specification.
- It MUST initialize HomeDomain with the AD DS domain name of the AD DS domain to which it is joined, if any.
- It MUST initialize DFSNamespacesList to the list of domain-based and standalone DFS namespaces that it hosts. This list can be obtained from DFS metadata (as specified in [\[MS-DFSNM\]](#)), a configuration file, a configuration store (such as the Windows registry), or from other implementation-defined means.

3.2.4 Higher-Layer Triggered Events

The server receives and acts upon DFS referral request messages. No other higher-layer triggered events are used.

3.2.5 Message Processing Events and Sequencing Rules

3.2.5.1 Receiving a DFS Referral Request

The server receives DFS referral requests from clients when they need to resolve DFS paths into file server paths.

The **MaxReferralLevel** field of the [SMB2_REQ_GET_DFS_REFERRAL \(section 2.2.2\)](#) message contains the highest DFS referral version understood by the client. The server MUST respond to a SMB2_REQ_GET_DFS_REFERRAL with a version number less than or equal to the value specified by the client. It SHOULD respond with the highest version number that it supports that can meet this constraint. [<25>](#)

The server MUST determine the maximum referral response size by the buffer size specified by the client in the TRANS2_GET_DFS_REFERRAL SMB subcommand, as specified in [\[MS-SMB\]](#) section 2.2.13.18. The server MUST restrict the referral response to the response buffer size. It MUST NOT return partial referral entries. If the response buffer size does not permit at least one complete referral entry, the server MUST fail the referral request with STATUS_BUFFER_OVERFLOW (0x80000005).

Processing of individual referral requests is explained in the following sections.

3.2.5.2 Receiving a Domain Referral Request

A server that is not running on a DC MUST fail this request with a STATUS_INVALID_PARAMETER (0xC000000D) return code.

3.2.5.3 Receiving a DC Referral Request

A server that is not running on a DC MUST fail this request with a STATUS_INVALID_PARAMETER (0xC000000D) return code.

3.2.5.4 Receiving a sysvol Referral Request

A server that is not running on a DC MUST fail this request with a STATUS_NO_SUCH_DEVICE (0xC000000E) return code.

3.2.5.5 Receiving a Root Referral Request or Link Referral Request

If the DFS namespace for which referral is sought, as identified by the second path component, is not present in DFSNamespaceList, the server MUST fail the referral request with a STATUS_NO_SUCH_DEVICE (0xC000000E) return code.

If the DFS referral request path has more than two components, the server MUST attempt to find a DFS link in the DFSNamespaceList, which is a complete prefix of the DFS referral request path. For example, if the DFS referral request path is "\MyServer\MyDfs\dir1\link1\dir2", then the DFS link "\MyServer\MyDfs\dir1\link1" is a complete prefix of the DFS referral request path. On the other hand, the same DFS link is not a complete prefix of the DFS referral request path "\MyServer\MyDfs\dir1\link2\dir2". The server MUST do the comparison in a case-insensitive manner, taking into account the case of a DFS referral request path whose first path component is in the fully qualified domain name form when the DFS root or DFS link's first path component is in the NetBIOS form (or vice versa).

If a DFS link that is a complete prefix of the DFS referral request path is identified, the server MUST return a DFS link referral response; otherwise, if it has a match for the DFS root, it MUST return a root referral response.

For a domain-based DFS namespace, the server MUST use the DFS metadata in the AD DS object of the DFS namespace (as specified in [\[MS-DFSNM\]](#) section 2.3) to respond to the DFS referral request. When the server maintains DFSMetadataCache, it MAY use the cached information to respond to referral requests. Some of the relevant fields in the DFS metadata for a domain-based DFS namespace are as follows:

- The DFS namespace Root **BLOB** contains DFS root information.
- The DFS namespace Link BLOB contains DFS Link information.
- The **Prefix** and **ShortPrefix** fields of a DFS RootOrLink ID BLOB MUST contain a DFS root or a DFS Link name.
- The **ReferralTTL** field of a DFS namespace Root BLOB or a DFS namespace Link BLOB MUST contain the time-out associated with the DFS root or link.
- The DFS Target List BLOB MUST contain DFS link target information.
- The **Type** field of a DFS RootOrLinkID BLOB of a DFS namespace Root BLOB MUST contain the DFS referral site costing setting for the DFS namespace.
- The **Type** field of a DFS RootLinkId BLOB of a DFS namespace Root BLOB or a DFS namespace Link BLOB MUST contain the DFS client target fallback and DFS in-site referral mode settings for the DFS root or DFS Link.
- The PKT_ENTRY_TYPE_OUTSIDE_MY_DOM bit of the **Type** field in a DFS RootOrLinkID BLOB of a DFS namespace Link BLOB MUST be set for a DFS interlink.
- When not used as a time stamp, the **TargetTimeStamp** field of a Target Entry BLOB MUST contain the DFS target priority information. The **PriorityClass** and **PriorityRank** fields in such a **TargetTimeStamp** field MUST contain the priority class and priority rank of the target respectively.

The server SHOULD do the following once it has identified the DFS root targets or DFS link targets that correspond to the DFS root or DFS link to be used for the DFS referral response.

- From the IP address of the client, determine the AD DS site of the client as specified in [\[MS-ADTS\]](#).
- Determine the AD DS site of each target. For a domain-based DFS namespace, the server SHOULD NOT use the Site Information BLOB in the DFS metadata. The AD DS site of a DFS target SHOULD instead be determined from the IP address of the DFS target as specified in [\[MS-ADTS\]](#). The server MUST use the hostname of the target—the first component in the path— to determine the IP address of the target. [<26>](#)
- If the server supports DFS referral site costing, and site costing is enabled for the DFS namespace, determine the AD DS site cost from the client's AD DS site to the target's AD DS site if they are not in the same AD DS site. If the client's AD DS site, the target's AD DS site, or the AD DS site cost between them cannot be determined, the cost assigned to that target is implementation specific. [<27>](#)

The server MUST sort the targets as follows when either the server does not support DFS target priority or all the targets have the default priority. For more information, see [\[MS-DFSNM\]](#) section 2.2.1.9.

- If DFS referral site costing is disabled or not supported, or all targets have the same default priority, sort the targets into two target sets: one target set MUST consist of targets in the same AD DS site as the client and the other target set MUST consist of all remaining targets.
- If DFS referral site costing is enabled, sort the targets into target sets of increasing AD DS site cost from the target to the client.

In both cases, within each target set, the server SHOULD randomize the order of the targets on a per-DFS response basis to enable load-sharing across the targets. If DFS in-site referral mode is enabled for the DFS namespace, the server SHOULD remove all targets that are not in the same AD DS site of the client. If the DFS in-site referral mode is enabled for the DFS root, the server MUST apply it for all the DFS link targets. The server MUST use the DFS in-site referral mode setting of each DFS link only when in-site mode is not enabled for the DFS root. [<28>](#)

A server that supports DFS referral version 4 MUST also support DFS target priority.

When DFS referral site costing is enabled for the DFS namespace or at least one DFS target has a non-default priority, the server SHOULD sort the targets in the DFS referral response as follows:

- At a coarse-grained level, lay out targets in the following order in terms of priority class. For more information, see [\[MS-DFSNM\]](#) section 2.2.1.9.
 - Targets in the DfsGlobalHighPriorityClass priority class. This will be referred to as group 1 for identification purposes.
 - Targets in the DfsSiteCostHighPriorityClass, DfsSiteCostNormalPriorityClass, and DfsSiteCostLowPriorityClass priority classes. This will be referred to as group 2 for identification purposes.
 - Targets in the DfsGlobalLowPriorityClass priority class. This will be referred to as group 3 for identification purposes.
- Sort targets within each group based on whether or not DFS referral site costing is enabled for the namespace, as specified earlier for the case of targets with default priorities. This results in an AD DS site cost based ordering of targets within each of the three groups.
- If DFS in-site referral mode is enabled for the DFS namespace, the server SHOULD remove all targets that are not in the same AD DS site of the client from group 2.
- Further sort targets having the same AD DS site cost in group 2 on the basis of priority class in the order of DfsSiteCostHighPriorityClass first, followed by DfsSiteCostNormalPriorityClass, and finally DfsSiteCostLowPriorityClass.
- Within each of the three groups, further sort targets that have the same AD DS site cost and priority class in order of decreasing priority rank, with 0 (0x0000) being the highest priority rank and 31 (0x001F) the lowest priority rank. The server SHOULD create target sets that consist of targets that have the same AD DS site cost, priority class, and priority rank.
- The server SHOULD randomly reorder targets in each target set on a per-DFS referral response basis, to enable load-sharing across the targets.

Because there are three dimensions available for the sorting of targets—AD DS site cost, priority class and priority rank—additional implementation defined sorting policies MAY be used within each of the three groups specified above. [<29>](#)

After the sorted list of targets is available, noting that the list may be empty due to DFS in-site referral mode, the server still MUST initialize the RESP_GET_DFS_REFERRAL referral header as follows:

- For a DFS root referral response, **PathConsumed** MUST be set to the length in bytes of the first two path components in the referral request. For a DFS Link referral response, **PathConsumed** MUST be set to the length in bytes of the DFS referral request path prefix that matches the DFS Link.
- **NumberOfReferrals** MUST be set to the number of complete DFS referral entries that can be returned in the response buffer provided by the DFS referral request. The server MAY silently drop targets that will not fit in the buffer. However, if the buffer size is insufficient to return even one referral entry, the server MUST fail the referral request, as specified in section [3.2.5.1.<30>](#)
- If DFS root targets are returned or if a DFS interlink is returned, the ReferralServers bit of the referral entry MUST be set to 1. In all other cases, it MUST be set to 0.
- If DFS root targets are returned or if a DFS interlink is returned, the StorageServers bit of the referral entry MUST be set to 1. In all other cases, it MUST be set to 0.
- For a DFS referral version 4 response, the **TargetFailback** field MUST be set to 0x0001 in a DFS root referral response if DFS client target failback is enabled for the DFS root. For link referrals, the **TargetFailback** field MUST be set to 0x0001 if DFS client target failback is enabled either for the link itself or for the DFS root. In all other cases, the **TargetFailback** field MUST be set to 0x0000.
- All other fields are reserved and SHOULD be set to 0. [<31>](#)

The server MUST add one referral entry structure for each target returned and initialize each referral entry as follows:

- The **VersionNumber** field MUST be set to the minimum of the highest DFS referral version supported by the server and the value specified in the **MaxReferralLevel** field of [SMB2_REQ_GET_DFS_REFERRAL](#).
- The **Size** field MUST be set to the total size of the referral entry in bytes, including any padding at the end of the structure.
- The **ServerType** field MUST be set to 0x0001 if root targets are returned. In all other cases, the **Size** field MUST be set to 0x0000.
- The **NameListReferral** field MUST be set to 0.
- For DFS referral version 4 responses, the TargetSetBoundary bit MUST be set to 1 for each target that is the first target in a target set, and MUST be set to 0 for all other targets.
- The **TimeToLive** field MUST be set to the time-out value associated with the DFS root or DFS Link.
- For a DFS root referral response:
 - The **DFSPathOffset** field MUST be set to the offset in bytes from the beginning of the referral entry to a string that contains the first two path components of the DFS referral request path.

- The **DFSAlternatePathOffset** field MUST be set as specified in section [2.2.4.3.1](#).
- For a link referral response:
 - The **DFSPathOffset** field MUST be set to the offset in bytes from the beginning of the referral entry to a string that contains the DFS referral request path prefix that matches a DFS Link.
 - The **DFSAlternatePathOffset** field MUST be set as specified in section [2.2.4.3.1](#).
- The **NetworkAddressOffset** field MUST be set to the offset in bytes from the beginning of the referral entry to a string that contains the DFS target for the entry.
- All other fields are reserved and MUST be set to 0.

Servers MAY return fully qualified DNS hostnames in responses to root referral request and link referral requests. [<32>](#)

3.2.6 Timer Events

No protocol timer events are required on the server.

3.2.7 Other Local Events

No other local events are used on the server.

3.3 Domain Controller Details

This section specifies how a DFS server on a DC processes and responds to DFS referral requests from DFS clients. It is applicable only to DFS servers running on AD DS DCs. [<33>](#)

A DFS server on a DC MUST respond to domain, DC, sysvol, root, and link-referral requests.

If that DFS server is also hosting DFS root targets, then in addition to the requirements in this section, it MUST also conform to the requirements in section [3.2](#) while processing referral requests for those DFS root targets.

3.3.1 Abstract Data Model

No abstract data model is required for the DC role.

3.3.2 Timers

No timers are required by the protocol.

3.3.3 Initialization

No initialization is required for the DC role.

3.3.4 Higher-Layer Triggered Events

Servers receive and act upon DFS referral requests. No other higher-layer triggered events are used.

3.3.5 Message Processing Events and Sequencing Rules

3.3.5.1 Receiving a DFS Referral Request

For more information about DFS referral requests, see section [3.2.5.1](#). The processing of individual referral requests is explained in the following sections.

3.3.5.2 Receiving a Domain Referral Request

The server MUST return a list of AD DS domains, for the AD DS forest in which the DC exists, in both NetBIOS and fully qualified domain name forms.

To process a domain referral, the client MUST support DFS referral version 3 or DFS referral version 4. If the requested DFS referral version is less than 3, the server SHOULD fail the referral with the STATUS_UNSUCCESSFUL (0xC0000001) return code. [<34>](#)

Because there is no difference between DFS referral versions 3 and 4 for a domain referral response, a server that supports DFS referral version 4 MAY return a DFS referral version 3 response even if the DFS client specified DFS referral version 4. [<35>](#)

When the number of AD DS domains is more than can fit in the response buffer and the response buffer is less than 56 KB, the following sequence of messages MUST occur:

- The server fails the referral request with STATUS_BUFFER_OVERFLOW (0x80000005).
- The client retries by doubling the buffer size up to a maximum of 56 KB.

Otherwise the server MUST fill the response buffer as follows:

- MUST return both the NetBIOS and fully qualified names of an AD DS domain or return no entry for the AD DS domain.
- MUST give preference to the NetBIOS and fully qualified names of the DC's own AD DS domain first, and then give preference to other AD DS domains. This ensures that a client in the same AD DS domain as the server will always be able to access SYSVOL and NETLOGON paths correctly. [<36>](#)
- MUST return only complete names, and not truncate a name to fit in the response buffer.
- MUST return the data equal to the requested buffer size or 56 KB of data, whichever is less.

The server MUST initialize the [RESP_GET_DFS_REFERRAL](#) referral header for the response as follows: [<37>](#)

- **PathConsumed** MUST be set to 0.
- **NumberOfReferrals** MUST be set to the number of complete referral entries that can fit in the response buffer provided by the DFS referral request. The server MAY discard excess referral response entries and MAY return a STATUS_BUFFER_OVERFLOW (0x80000005) error if the response buffer is too small.
- The **ReferralServers** and **StorageServers** fields are set to 0.

The server MUST add one referral entry structure for each [AD DS domain name](#) returned and initialize each referral entry as follows:

- The **VersionNumber** field MUST be set to 0x0003.

- The **Size** field MUST be set to the total size of the referral entry in bytes, including any padding at the end of the structure.
- The **NameListReferral** field MUST be set to 1.
- The **TimeToLive** field MAY be set to a non-zero cache time-out value in seconds. [<38>](#)
- The **SpecialNameOffset** field MUST be set to the offset in bytes from the beginning of the referral entry to the string that contains the AD DS domain name for the entry.
- All other fields SHOULD be set to 0.

3.3.5.3 Receiving a DC Referral Request

The AD DS domain name in the referral request MUST be either an AD DS domain in the current forest or an AD DS domain in a trusted forest. The server MUST fail DC referral requests for other AD DS domain names with a STATUS_INVALID_PARAMETER (0xC000000D) return code.

DC referrals require that the DFS client support DFS referral version 3 or version 4. If the requested DFS referral version is less than 3, the server SHOULD fail the referral with the STATUS_UNSUCCESSFUL (0xC0000001) return code. [<39>](#)

Because there is no difference between DFS referral versions 3 and 4 for a DC referral response, a server that supports DFS referral version 4 MAY return a DFS referral version 3 response even if the DFS client specified DFS referral version 4. [<40>](#)

For a valid [AD DS domain name](#), the server MUST return as many complete DC names as can fit in the response buffer. The format of the DC names returned MUST correspond to the format of the AD DS domain name in the referral request.

- If the request contains a fully qualified AD DS domain name, the response MUST contain a fully qualified DC hostname.
- If the request contains a NetBIOS AD DS domain name, the response MUST contain a NetBIOS DC hostname.

The server MUST initialize the [RESP_GET_DFS_REFERRAL](#) referral header for the response as follows:

- The **PathConsumed** field MUST be set to 0.
- The **NumberOfReferrals** field MUST be set to 1, independent of the number of DC names returned.
- The ReferralServers and StorageServers bits MUST be set to 0.

The server MUST place exactly one referral entry structure in the referral response. The server MUST initialize this referral entry as follows:

- The **VersionNumber** field MUST be set to 0x0003.
- The **Size** field MUST be set to the total size of the referral entry in bytes, including any padding at the end of the structure.
- The **NameListReferral** field MUST be set to 1.
- The **TimeToLive** field MAY be set to a non-zero cache time-out value in seconds. [<41>](#)

- The **SpecialNameOffset** field MUST be set to the offset in bytes from the beginning of the referral entry to the string that contains the AD DS domain name for the referral response.
- The **NumberOfExpandedNames** field MUST be set to the number of DCs returned.
- The **ExpandedNameOffset** field MUST be set to the offset in bytes from the beginning of the referral entry to the first null-terminated DC Unicode string. Each DC name immediately follows its null-terminated predecessor without any padding. An implementation MUST use the value in the **NumberOfExpandedNames** field to determine how many names are present in the list at **ExpandedNameOffset**.
- All other fields SHOULD be set to 0.

3.3.5.4 Receiving a sysvol Referral Request

The [AD DS domain name](#) in the referral request MUST be either an AD DS domain in the current forest or an AD DS domain in another trusted forest. The DFS server MUST fail sysvol referral requests with other AD DS domain names with a STATUS_NOT_FOUND (0xC0000225) return code.

Sysvol referrals require that the DFS client support DFS referral version 3 or version 4. If the requested DFS referral version is less than 3, the server SHOULD fail the referral with the STATUS_UNSUCCESSFUL (0xC0000001) return code. [<42>](#)

The DFS server MUST determine the list of DCs for the AD DS domain name specified in the sysvol referral request from AD DS and then generate a list of DCs to return in the referral response that MUST be in either of the following forms.

- "\\<dcname>\SYSVOL" if the second component of referral request path is "SYSVOL".
- "\\<dcname>\NETLOGON" if the second component of referral request path is "NETLOGON".

Where <dcname> is a DC in the AD DS domain specified in the first path component of the referral request.

The format of the <dcname> returned MUST correspond to the format of the AD DS domain name in the referral request.

- If the request contains a fully qualified AD DS domain name, the response MUST contain fully qualified DC hostnames.
- If the request contains a NetBIOS AD DS domain name, the response MUST contain NetBIOS DC hostnames.

A server MUST fail referral requests with more than two path components with a STATUS_NOT_FOUND (0xC0000225) return code.

Servers MAY support DFS referral site costing for sysvol referral requests. [<43>](#)

After the candidate list of targets is available, the server SHOULD do the following:

- From the IP address of the client, determine the AD DS site of the client, as specified in [\[MS-ADTS\]](#).
- From the first path component of a target (the hostname), determine its AD DS site.
- If the server supports DFS referral site costing, determine the AD DS site cost from the client AD DS site to the target's AD DS site if they are not in the same AD DS site. If the client AD DS site,

the target AD DS site, or the AD DS site cost between them cannot be determined, the cost assigned to that target is implementation-specific. [<44>](#)

The server MUST sort the targets as specified in the following:

- If DFS referral site costing is disabled, sort the targets into two target sets: one target set MUST consist of targets in the same AD DS site as the client and the other target set MUST consist of all remaining targets.
- If DFS referral site costing is enabled, sort the targets into target sets of increasing AD DS site cost from the target to the client.

In both cases, the server SHOULD randomize the targets within each target set, to enable load-sharing across the targets. [<45>](#)

The server MUST initialize the [RESP_GET_DFS_REFERRAL](#) referral header as follows for the response.

- The **PathConsumed** field MUST be set to length in bytes of the DFS referral request path.
- The **NumberOfReferrals** field MUST be set to the number of complete DFS referral entries that can be returned in the response buffer provided by the DFS referral request. The server MAY silently drop targets that will not fit in the buffer. However, if the buffer size is insufficient to return even one referral entry, the server MUST fail the request with STATUS_BUFFER_OVERFLOW (0x80000005).
- The **ReferralServers** field MUST be set to 0 and the **StorageServers** field MUST be set to 1.
- For DFS referral version 3 responses, the **TargetFallback** field MUST be set to 0. For DFS referral version 4 responses, the **TargetFallback** field MAY be set to 1. [<46>](#)
- All other fields SHOULD be set to 0.

The server MUST add one referral entry structure for each target returned. The server MUST initialize each referral entry as follows:

- The **VersionNumber** field MUST be set to the minimum of the highest DFS referral version supported by the server and the value specified in the **MaxReferralLevel** field of SMB2_REQ_GET_DFS_REFERRAL.
- The **Size** field MUST be set to the total size of the referral entry in bytes, including any padding at the end of the structure.
- The **NameListReferral** field MUST be set to 0.
- For a version 4 DFS referral response, the **TargetSetBoundary** MUST be set to 1 if the target is the first target of a target set, and MUST be set to 0 if the target is other than the first target of a target set.
- The **TimeToLive** field SHOULD be set to a non-zero time-out value in seconds. [<47>](#)
- The **DFSPathOffset** field MUST be set to the offset in bytes from the beginning of the referral entry to the string that contains the DFS referral request path.
- The **DFSAlternatePathOffset** field MUST be set to the offset in bytes from the beginning of the referral entry to the string that contains the DFS referral request path.

- The **NetworkAddressOffset** field MUST be set to the offset in bytes from the beginning of the referral entry to the string that contains the target.
- All other fields MUST be set to 0.

3.3.5.5 Receiving a Root Referral Request or Link Referral Request

Root referral requests and link referral requests MUST be handled as specified in section [3.2.5.5](#), with the following exception: a DC MUST process root referral requests and link referral requests for domain-based DFS namespaces in the AD DS domain for which it is a DC. The name of the DFS namespace in the second component of the DFS referral request path MUST be validated against the domain-based DFS namespaces in the AD DS domain of the DC. This MUST be done by searching the name attribute in the AD DS objects under the DFS configuration container of the AD DS domain.

3.3.6 Timer Events

No protocol timer events are required on the server.

3.3.7 Other Local Events

No local events are required on the server.

4 Protocol Examples

The following sections describe several operations as used in common scenarios to illustrate the function of the Distributed File System (DFS): Referral Protocol.

4.1 Domain Referral

The following sequence diagram shows a domain referral.

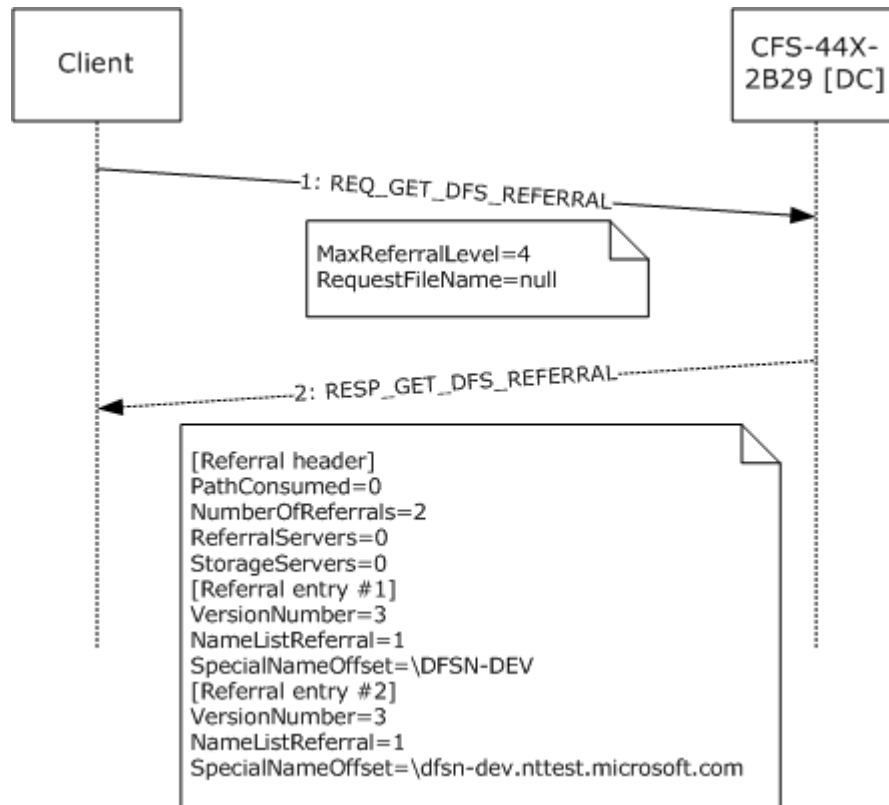


Figure 8: Domain referral sequence

1. The client sends a [SMB2_REQ_GET_DFS_REFERRAL \(section 2.2.2\)](#) to CFS-44X-2B29, a DC in the DFSN-DEV.NTTEST.MICROSOFT.COM AD DS domain. The highest DFS referral version understood by the client is 4, as indicated by the **MaxReferralLevel** field of SMB2_REQ_GET_DFS_REFERRAL. The DFS path specified in the DFS referral request is a 0-length string, indicating a domain referral. To illustrate the effect of a client specifying a referral version that is later than what the server supports, the client's referral version is shown as 4 in this example.
2. The DC's response has two referral entries, PathConsumed is 0, ReferralServers and StorageServers bits are set to 0. Each referral entry has the **VersionNumber** field set to 3. While the client's referral version in the request was 4, the server has replied with version 3. The NameListReferral bit is set to 1 in each referral entry. The strings pointed to by the **SpecialNameOffset** field of the individual referral entries are AD DS domain names.

4.2 DC Referral

The following sequence diagram shows a DC referral.

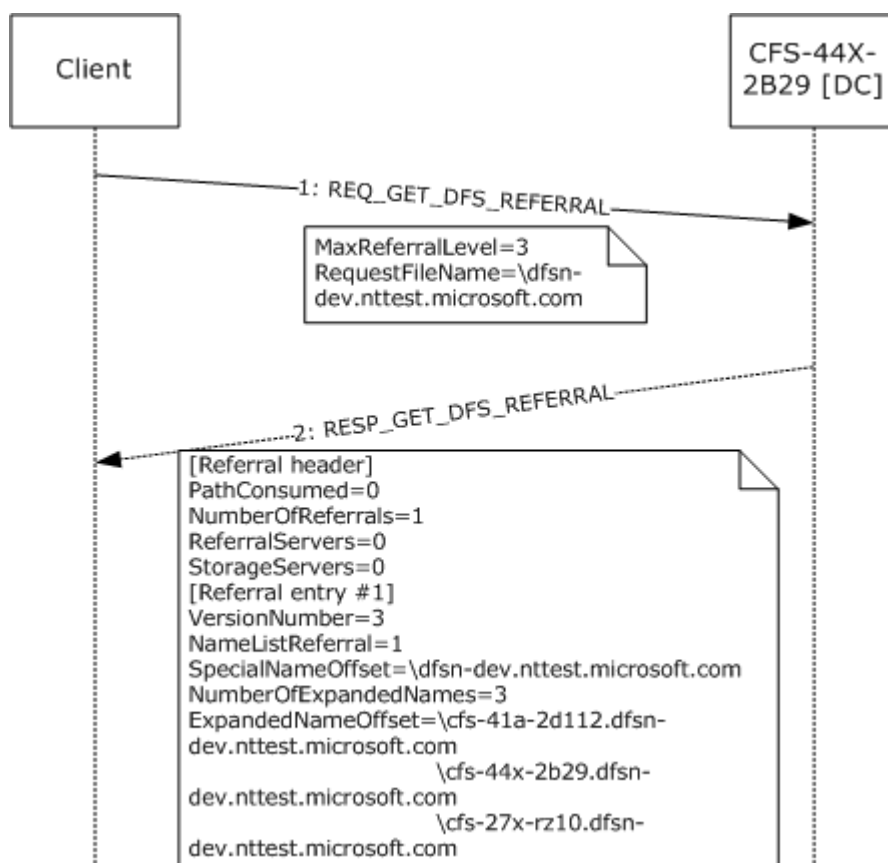


Figure 9: Domain controller referral

1. The client sends a [SMB2_REQ_GET_DFS_REFERRAL \(section 2.2.2\)](#) to CFS-44X-2B29, a DC in the DFSN-DEV AD DS domain. The highest DFS referral version understood by the client is 3, as indicated by the **MaxReferralLevel** field of SMB2_REQ_GET_DFS_REFERRAL. The DFS path specified in the DFS referral request is the string \dfs-dev.nttest.microsoft.com, indicating a DC referral.
2. The DC's response has one referral entry, PathConsumed is 0, ReferralServers and StorageServers bits are set to 0. The referral entry has the **VersionNumber** field set to 3. The NameListReferral bit is set to 1 in the referral entry. The string pointed to by the **SpecialNameOffset** field of the referral entry is the AD DS domain name for which the DC list returned. The **NumberOfExpandedNames** field is set to 3, indicating that there are three DCs in the DC list returned. The **ExpandedNameOffset** field in the referral entry points to a sequence of three strings, each of which is a DC name. The DC names returned are in fully qualified domain name format, because the AD DS domain name in the request is also in the fully qualified domain name format.

4.3 Domain-Based DFS Root Referral

The following sequence diagram shows a root referral for a domain-based DFS namespace.

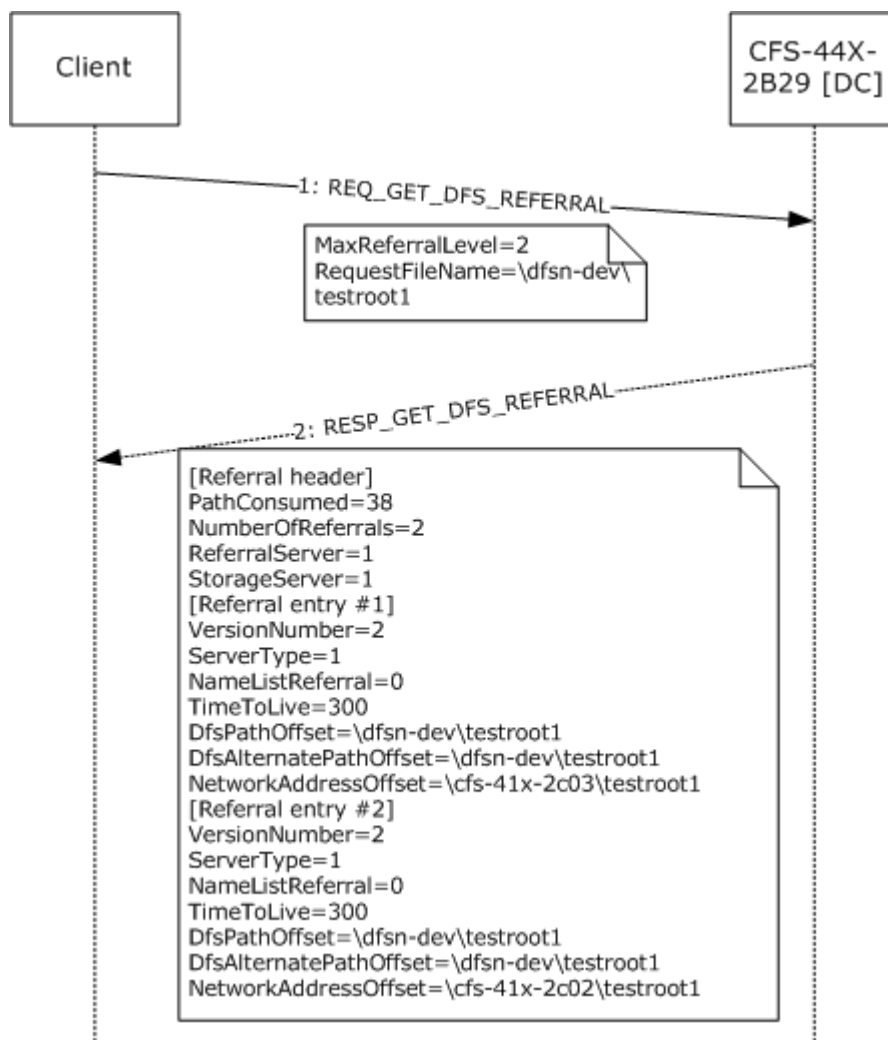


Figure 10: Domain-based DFS root referral

1. The client sends a [SMB2_REQ_GET_DFS_REFERRAL \(section 2.2.2\)](#) to CFS-44X-2B29, a DC in the DFSN-DEV AD DS domain. The highest DFS referral version understood by the client is 2, as indicated by the **MaxReferralLevel** field of SMB2_REQ_GET_DFS_REFERRAL. The DFS path specified in the DFS referral request is \dfs-dev\testroot1, a domain-based DFS namespace in the DFSN-DEV domain.
2. The DC's referral response contains two referral entries, each having referral version 2. The ReferralServers and StorageServers bits are set to 1 and the **ServerType** field is set to 1 in the response since DFS root targets are returned. The strings pointed to by the **NetworkAddressOffset** field of the individual referral entries are the DFS root targets for the domain-based DFS namespace.

4.4 Domain-Based DFS Link Referral

The following sequence diagram shows a DFS link referral for a domain-based DFS namespace.

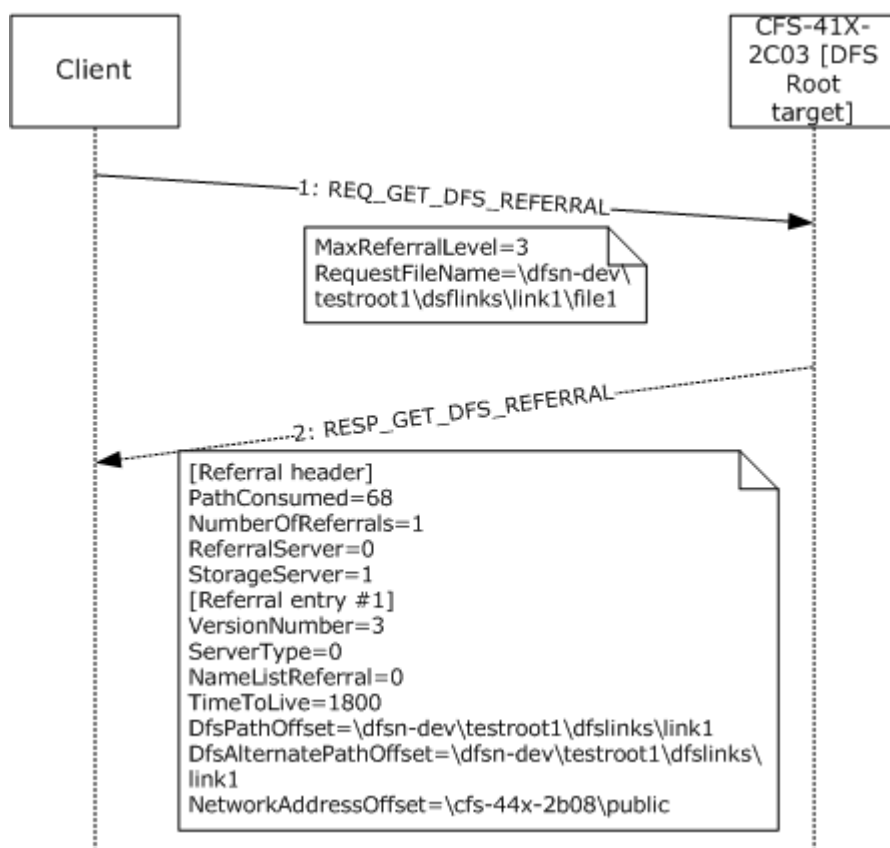


Figure 11: Domain-Based DFS Link Referral

1. The client sends a [SMB2_REQ_GET_DFS_REFERRAL \(section 2.2.2\)](#) to CFS-41X-2C03, a DFS root target for the domain-based DFS namespace \\dfs-dev\testroot1 in the DFSN-DEV AD DS domain. The highest DFS referral version understood by the client is 3, as indicated by the **MaxReferralLevel** field of SMB2_REQ_GET_DFS_REFERRAL. The DFS path specified in the DFS referral request is \dfs-dev\testroot1\dfslinks\link1\file1, a path to a file in the DFS namespace.
2. The DFS root target's referral response contains one referral entry having referral version 3. The ReferralServers bit is set to 0 and StorageServers bit is set to 1 indicating that the target is not another DFS root. The **ServerType** field is set to 0 in the response because a DFS link target is returned. The string pointed to by the **NetworkAddressOffset** field of the referral entry is the DFS link target.

4.5 Domain-Based DFS Root Referral Packet Trace

The following figure shows the network packet trace of a domain-based DFS root referral request. The [REQ_DFS_GET_REFERRAL](#) structure is at offset 0x7E in the hexadecimal dump shown. **MaxReferralLevel** field is at offset 0x7E (=3) and **RequestFileName** is at offset 0x80 (=\\dfs-dev\testroot1).

```

ETHERNET: EType - Internet IP (IPv4)
IP: Protocol = TCP - Transmission control; Packet ID - 48186; Total IP
Length = 154; Options = NO Options
TCP: Control Bits: .AP..., len: 114, seq:1634334697-1634334811,
ack:2251252280, win:64498, src: 2086 dst: 139 (NBT Session)
NBT: SS: Session Message, Len: 110
SMB: C transact2 NT Get DFS Referral
    SMB: Command = C transact2
        SMB: Word count = 15
        SMB: Total param count = 42
        SMB: Total data count = 0
        SMB: Max param count = 0
        SMB: Max data count = 4096
        SMB: Max setup words = 0 (0x0)
        SMB: Transact Flags Summary = 0 (0x0)
            SMB:.....0 = Leave session intact
            SMB:.....0. = Response required
        SMB: Transact timeout = 0 (0x0)
        SMB: Parameter count = 42 (0x2A)
        SMB: Parameter offset = 68 (0x44)
        SMB: Data count = 0 (0x0)
        SMB: Data offset = 0 (0x0)
        SMB: Setup count = 1
        SMB: Setup words
        SMB: Transact2 function = TRANS2_DFS_GET_REFERRAL
        SMB: Byte count = 45
        SMB: Parameter bytes
        SMB: Transaction parameters
            SMB: DFS Max Referral Level = 3 (0x3)
            SMB: DFS Request Filename = \dfs-dev\testroot1
00000: 00 D0 04 3C D8 00 00 B0 D0 C2 01 3C 08 00 45 00
00010: 00 9A BC 3A 40 00 80 06 76 EF 9D 3B FD 44 0A C1
00020: 21 F3 08 26 00 8B 61 69 F7 E9 86 2F 62 38 50 18
00030: FB F2 C7 C0 00 00 00 00 00 00 6E FF 53 4D 42 32 00
00040: 00 00 00 18 07 C8 00 00 00 00 00 00 00 00 00 00
00050: 00 00 02 20 B8 0B 00 38 40 01 0F 2A 00 00 00 00
00060: 00 00 10 00 00 00 00 00 00 00 00 00 00 00 2A 00 44
00070: 00 00 00 00 00 01 00 10 00 2D 00 00 00 00 03 00
00080: 5C 00 64 00 66 00 73 00 6E 00 2D 00 64 00 65 00
00090: 76 00 5C 00 74 00 65 00 73 00 74 00 72 00 6F 00
000A0: 6F 00 74 00 31 00 00

```

Figure 12: REQ_DFS_GET_REFERRAL packet trace

The following figure shows the network packet dump of a domain-based DFS root referral response. The referral response consists of the DFS referral header, [RESP_GET_DFS_REFERRAL](#) at offset 0x72 in the hexadecimal dump. The **PathConsumed** field is at offset 0x72 (=0x26) and the **NumberOfReferrals** field is at offset 0x74 (=2). The first referral entry, of type [DFS_REFERRAL_V3](#), starts at 0x7A and the second referral entry, also of type [DFS_REFERRAL_V3](#), starts at offset 0x9C. The referral header and the first referral entry have been marked in the hexadecimal dump for identification purposes. The hexadecimal dump shows that the two referral entries are immediately after one another and all the strings referenced in the referral entries are at the end of all the referral entries, starting at offset 0xBE in the hexadecimal dump.

The **NetworkAddressOffset** field of the first referral entry, containing the offset of the DFS root target from the start of the referral entry, is at offset 0x8A. It contains 0x94. The first referral entry is at offset 0x7A. Hence, the DFS root target that corresponds to the first referral entry is at offset 0x7A+0x94=0x10E. The null-terminated Unicode string at offset 0x10E in the hexadecimal dump is \cfs-41x-2c03\testroot1, one of the DFS root targets of the domain-based DFS namespace \\dfs-dev\testroot1.

```

ETHERNET: EType - Internet IP (IPv4)
IP: Protocol - TCP - Transmission control; Packet ID = 12850; Total IP Length - 352; Options = NO Options
TCP: control Bits: .AP..., len: 312, seq:2251252280-2251252592, ack:1634334811, win:64482,
src: 139 (NBT session) dst: 2086
NBT: SS: Session Message, Len: 308
SMB: R transact2 NT Get DFS Referral (response to frame 48)
  SMB: Command = C transact2
    SMB: Word count = 10
    SMB: Total parm count = 0
    SMB: Total data count = 252
    SMB: Parameter count = 0 (0x0)
    SMB: Parameter offset = 56 (0x38)
    SMB: Parameter Displacement = 0 (0x0)
    SMB: Data count = 252 (0x0FC)
    SMB: Data offset = 56 (0x38)
    SMB: Data Displacement = {0x0}
    SMB: Setup count = 0
    SMB: Byte Count = 253
    SMB: Byte parameters
    SMB: Transaction data
      SMB: DFS Path consumed = 38 (0x26)
      SMB: DFS Number of Referrals = 2 (0x2)
      SMB: DFS Server Function = 3 (0x3)
        SMB: .....1 = Referral Server
        SMB: .....1. = Storage Server
      SMB: DFS version 3 Referral
        SMB: DFS Version Number = 3 (0x3)
        SMB: DFS Server Type = SMB Server
        SMB: DFS TimeToLive = 300 (0x12C)
        SMB: DFS Filename = \dfs-dev\testroot1
        SMB: DFS 8.3 Filename = \dfs-dev\testroot1
        SMB: DFS Sharename = \cfs-41x-2c03\testroot1
      SMB: DFS Servicesite GUID = 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
      SMB: DFS version 3 Referral
        SMB: DFS Version Number = 3 (0x3)
        SMB: DFS Server Type = SMB Server
        SMB: DFS TimeToLive = 300 (0x12C)
        SMB: DFS Filename = \dfs-dev\testroot1
        SMB: DFS 8.3 Filename = \dfs-d\testroot1
        SMB: DFS Sharename = \cfs-41x-2c02\testroot1
      SMB: DFS Servicesite GUID = 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0000: 00 B0 D0 C2 01 3C 00 D0 04 3C 08 00 08 00 45 00
0001: 01 60 32 32 40 00 7E 06 02 32 0A C1 21 F3 9D 3B
0002: FD 44 00 88 08 26 86 2F 62 38 61 69 F8 5B 50 18
0003: FB E2 38 87 00 00 00 01 34 FF 53 4D 42 32 00
0004: 00 00 00 98 07 C8 00 00 00 00 00 00 00 00 00
0005: 00 00 02 20 B8 0B 00 38 40 01 0A 00 00 FC 00 00
0006: 00 00 38 00 00 00 FC 00 38 00 00 00 00 00 FD
0007: 00 00 26 00 02 00 03 00 00 00 03 00 22 00 01 00
0008: 00 00 2C 01 00 00 44 00 6C 00 94 00 00 00 00 00
0009: 00 00 00 00 00 00 00 00 00 00 00 00 03 00 22 00
000A: 01 00 00 00 2C 01 00 00 22 00 4A 00 02 00 00 00
000B: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 5C 00
000C: 64 00 66 00 73 00 6E 00 2D 00 64 00 65 00 76 00
000D: 5C 00 74 00 65 00 73 00 74 00 72 00 6F 00 6F 00
000E: 74 00 31 00 00 00 5C 00 64 00 66 00 73 00 6E 00
000F: 2D 00 64 00 65 00 76 00 5C 00 74 00 65 00 73 00
0010: 74 00 72 00 6F 00 6F 00 74 00 31 00 00 00 5C 00
0011: 63 00 66 00 73 00 2D 00 34 00 31 00 78 00 2D 00
0012: 32 00 63 00 30 00 33 00 5C 00 74 00 65 00 73 00
0013: 74 00 72 00 6F 00 6F 00 74 00 31 00 00 00 5C 00
0014: 63 00 66 00 73 00 2D 00 34 00 31 00 78 00 2D 00
0015: 32 00 63 00 30 00 32 00 5C 00 74 00 65 00 73 00
0016: 74 00 72 00 6F 00 6F 00 74 00 31 00 00 00

```

Figure 13: RESP_GET_DFS_REFERRAL packet trace

4.6 Standalone DFS Root Referral

The following sequence diagram shows a root referral for a standalone DFS namespace.

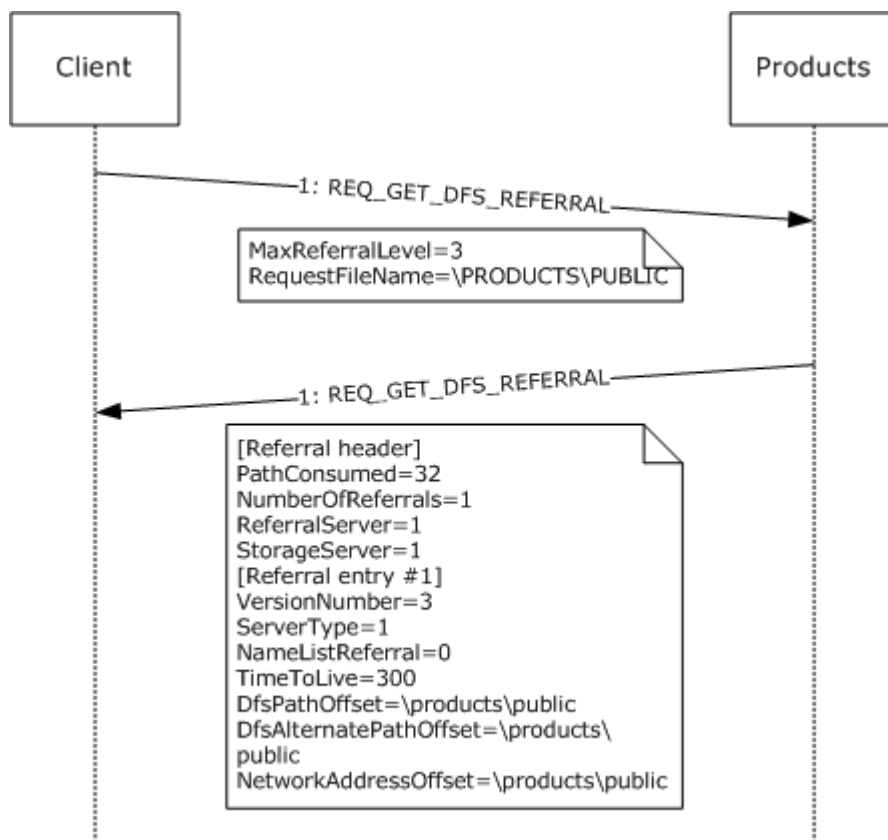


Figure 14: DFS root referral between client and server

The client sends a [SMB2_REQ_GET_DFS_REFERRAL](#) to PRODUCTS. The latest DFS referral version understood by the client is 3, as indicated by the **MaxReferralLevel** field of SMB2_REQ_GET_DFS_REFERRAL. The DFS path specified in the DFS referral request is \\PRODUCTS\\PUBLIC, a standalone DFS namespace.

The referral response contains one referral entry having a referral version 3. The ReferralServers and StorageServers bits are set to 1, and the **ServerType** field is set to 1 in the response since DFS root targets are returned. The strings pointed to by the **NetworkAddressOffset** field of the referral entry is the DFS root target for the standalone DFS namespace.

5 Security

The following sections specify security considerations for implementers of the Distributed File System (DFS): Referral Protocol.

5.1 Security Considerations for Implementers

The Distributed File System (DFS): Referral Protocol operates within an authenticated SMB session. DFS referral servers SHOULD use the identity of the session to perform access checks on referral requests.

5.2 Index of Security Parameters

The Distributed File System (DFS): Referral Protocol defines no security parameters beyond its SMB transport.

6 Appendix A: Windows Behavior

The information in this specification is applicable to the following versions of Windows:

- Windows Server 2008
- Windows Server 2003
- Windows NT Workstation 4.0 SP2 and later
- Windows Vista
- Windows XP
- Windows 2000
- Windows Me
- Windows 98
- Windows 95 with Active Directory client

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Windows behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that Windows does not follow the prescription.

[<1> Section 1.8:](#) Windows uses only the values specified in [\[MS-ERREF\]](#).

[<2> Section 2.2.3:](#) Windows Server 2003 and Windows 2000 Server will add up to 192 bytes of padding after [DFS_REFERRAL_V3](#) entries. Windows Server 2008 and Windows Vista will not add any data after [DFS_REFERRAL_V3](#) structures.

[<3> Section 2.2.4:](#) Strings for referral entries created by Windows servers (Windows NT Workstation 4.0 SP2 and later, Windows 2000 Server, Windows Server 2003, and (Windows Server 2008) are always located after the last referral entry.

[<4> Section 2.2.4.2:](#) Windows 2000 and earlier set the **DFSAlternatePathOffset** to point to an 8.3 string representation of the string pointed to by **DFSPathOffset** if it is not a legal 8.3 string. Otherwise, if **DFSPathOffset** points to a legal 8.3 string, **DFSAlternatePathOffset** points to a separate copy of the same string. Windows Server 2003 and Windows Server 2008 set the **DFSPathOffset** and **DFSAlternatePathOffset** fields to point to separate copies of the identical string.

[<5> Section 2.2.4.3.1:](#) Windows 2000 and earlier set the **DFSAlternatePathOffset** field to point to an 8.3 string representation of the string pointed to by **DFSPathOffset** if it is not a legal 8.3 string. Otherwise, if **DFSPathOffset** points to a legal 8.3 string, **DFSAlternatePathOffset** points to a separate copy of the same string. Windows Server 2003 and Windows Server 2008 set the **DFSPathOffset** and **DFSAlternatePathOffset** to point to separate copies of the identical string.

[<6> Section 3.1:](#) DFS client functionality is available in the following Windows versions:

- Windows 95 with **Active Directory** client
- Windows 98
- Windows NT Workstation 4.0 SP2 and later

- Windows Me
- Windows 2000
- Windows XP
- Windows Server 2003
- Windows Vista
- Windows Server 2008
- The buffer size used by Windows DFS clients for all DFS referral requests (domain, DC, DFS root, DFS Link and SYSVOL) is 8 KB. Windows DFS clients retry on STATUS_BUFFER_OVERFLOW (0x80000005) by doubling the buffer size up to a maximum of 56 KB.

[<7> Section 3.1.1:](#) Windows clients cache information from referral requests to reduce network traffic and reduce file access latencies.

[<8> Section 3.1.1:](#) The default **TimeToLive** value for root referrals is 300 seconds and the default **TimeToLive** value for link referrals is 1800 seconds.

[<9> Section 3.1.2:](#) Windows clients (Windows 2000, Windows XP, Windows Server 2003, and Windows Server 2008) use 15 minutes as the BootstrapDCTimer period. The interval is configurable in Windows.

[<10> Section 3.1.4.1:](#) returns errors as follows: For root referral failures, if the DFS path is a domain path the error returned is STATUS_OBJECT_PATH_NOT_FOUND when talking to a DC and STATUS_NO_SUCH_DEVICE when talking to non-DC. If the DFS path is a non-domain path, then the error is STATUS_BAD_NETWORK_PATH or one of the DfscFailoverErrorList error codes: STATUS_DEVICE_OFF_LINE, STATUS_NO_SUCH_DEVICE, STATUS_NO_SUCH_LOGON_SESSION, STATUS_INSUFFICIENT_RESOURCES, STATUS_IO_TIMEOUT, STATUS_REMOTE_NOT_LISTENING, STATUS_DUPLICATE_NAME, STATUS_BAD_NETWORK_PATH, STATUS_NETWORK_BUSY, STATUS_INVALID_NETWORK_RESPONSE, STATUS_UNEXPECTED_NETWORK_ERROR, STATUS_NETWORK_NAME_DELETED, STATUS_BAD_NETWORK_NAME, STATUS_SHARING_PAUSED, STATUS_REQUEST_NOT_ACCEPTED, STATUS_VIRTUAL_CIRCUIT_CLOSED, STATUS_UNMAPPABLE_CHARACTER, STATUS_DISK_OPERATION_FAILED, STATUS_NETLOGON_NOT_STARTED, STATUS_USER_SESSION_DELETED, STATUS_CONNECTION_DISCONNECTED, STATUS_CONNECTION_REFUSED, STATUS_NETWORK_UNREACHABLE, STATUS_HOST_UNREACHABLE, STATUS_DFS_UNAVAILABLE. For link referral failures, the error codes returned are STATUS_BAD_NETWORK_PATH, STATUS_OBJECT_PATH_NOT_FOUND, and STATUS_INVALID_PARAMETER.

[<11> Section 3.1.4.1:](#) In case of an I/O operation, the error code returned by Windows is one of the DfscFailoverErrorList error codes in the previous list.

[<12> Section 3.1.4.1:](#) The error code returned is one of the DfscFailoverErrorList error codes previously listed.

[<13> Section 3.1.4.1:](#) Windows Server 2008 and Windows Vista DFS clients use this flag to avoid sending DFS referral requests needed to determine whether or not a path is in a DFS namespace; DFS clients in earlier Windows versions always send a DFS referral request to determine whether or not a path is in a DFS namespace.

[<14> Section 3.1.5.1:](#) Windows NT 4.0 and later versions issue DFS link referral requests to the DFS root target server. Windows 98 and Windows 95 DFS clients issue DFS link referral requests to DCs when accessing a domain-based DFS namespace.

<15> [Section 3.1.5.1:](#) The highest DFS referral version supported by the various Windows clients is as follows:

Windows client version	Highest DFS referral version
Windows Me Windows 95, and Windows 98, with QFE	3
Windows NT 4.0 SP5 and earlier	2
Windows NT 4.0 SP6 and later	3
Windows 2000 Server and Windows 2000	3
Windows XP SP2 and Windows XP SP1	3
Windows XP with QFE	4
Windows Server 2003 SP1	3
Windows Server 2003 SP1 with QFE and later versions	4

DFS referral version 1 was not used in production versions of Windows.

<16> [Section 3.1.5.2:](#) Windows Server 2008 and Windows Server 2003 support, at most, one domain-based DFS namespace as a DFS link target. Such a link cannot have any other targets. A DFS link with more than one standalone DFS namespace or more than one share is also supported. DFS client target failover is not supported when an I/O operation issued to a DFS link target, which is a standalone DFS namespace, fails even if there is more than one target for the DFS link.

<17> [Section 3.1.5.4:](#) Windows clients retry the referral request with a bigger buffer size, as specified in [3.1](#).

<18> [Section 3.1.5.4.1:](#) Windows clients do not remove an existing [AD DS domain name](#) from DomainCache.

<19> [Section 3.1.5.4.3:](#) The Windows client returns STATUS_OBJECT_PATH_NOT_FOUND.

<20> [Section 3.1.5.4.3:](#) For I/O operations which have an [AD DS domain name](#) as the first component of the path, AD DS domain-joined Windows clients fail the original I/O operation when the root referral request to all the DCs for the AD DS domain fails. The Windows client returns STATUS_OBJECT_PATH_NOT_FOUND.

<21> [Section 3.1.5.4.3:](#) For I/O operations which have an AD DS domain name as the first component of the path, AD DS domain-joined Windows clients fail the original I/O operation and return STATUS_OBJECT_PATH_NOT_FOUND.

<22> [Section 3.1.5.4.3:](#) The Windows client handles this response in the same way it handles link referrals that it receives in response to link referral requests.

<23> [Section 3.1.5.4.3:](#) Windows Server 2003 and Windows XP DFS clients fail the I/O operation if ReferralCache entry refresh fails. Windows Server 2008 and Windows Vista DFS clients use a soft time-out equal to the TTL and a hard time-out equal to 4*TTL.

<24> [Section 3.2:](#) DFS server functionality is available only in the following Windows versions:

- Windows Server 2008
- Windows Server 2003

- Windows 2000 Server
- Windows NT Server 4.0 SP2 and later

Windows client operating systems, such as Windows NT 4.0 Professional, Windows Vista, Windows Me, Windows XP, Windows 2000 Professional, Windows 98 or Windows 95 do not implement the DFS server functionality.

[<25> Section 3.2.5.1:](#) The highest DFS referral version supported by each version of Windows on which DFS server functionality is implemented, is as follows:

Windows server version	Highest DFS referral version
Windows Server 2008	4
Windows Server 2003 SP1	4
Windows Server 2003	3
Windows 2000 Server	3
Windows NT 4.0	2

[<26> Section 3.2.5.5:](#) For historical reasons, Windows 2000 Server uses the Site Information BLOB in the DFS metadata of a domain-based DFS namespace to determine the AD DS site of a DFS target. Windows Server 2008 and Windows Server 2003 use the IP address of a DFS target to determine its AD DS site.

For interlinks, if the target is a domain-based DFS namespace, there cannot be more than one target for the link. Multiple standalone DFS namespace targets are supported as interlink targets. Therefore, there is no special handling for interlink targets. Referral response returns only the immediate interlink targets, not the targets of the domain-based DFS namespace interlink target itself.

[<27> Section 3.2.5.5:](#) Windows servers assign infinite cost to a target if it cannot identify either the target AD DS site or the target AD DS site cost from the client AD DS site.

Windows 2000 Server does not support DFS referral site costing. Windows Server 2008 and Windows Server 2003 support DFS referral site costing.

[<28> Section 3.2.5.5:](#) DFS in-site referral mode is applied only to DFS link targets and not to DFS root targets in Windows 2000 Server. DFS in-site referral mode is applied to all targets in Windows Server 2008 and Windows Server 2003.

Windows 2000 Server does not support DFS referral site costing. Only Windows Server 2008 and Windows Server 2003 support DFS referral site costing.

Windows servers randomize the order of targets within each target set to spread client load across the targets on a per DFS referral response basis.

[<29> Section 3.2.5.5:](#) Windows Server 2008, Windows Server 2003 SP1, and WS-Management sort targets having non-default priorities as described above.

[<30> Section 3.2.5.5:](#) Windows will silently drop targets that do not fit in the response buffer.

[<31> Section 3.2.5.5:](#) Windows Server 2008 and Windows Server 2003 SP1 set all other fields to 0. Earlier versions of Windows allow the other fields to be any value.

<32> [Section 3.2.5.5:](#) By default, all Windows servers return DNS hostnames that are not fully qualified for targets. However, Windows servers can be configured to return fully qualified DNS hostnames of targets.

<33> [Section 3.3:](#) AD DS DCs are supported only on Windows Server 2008, Windows Server 2003, and Windows 2000 Server.

<34> [Section 3.3.5.2:](#) Windows Server 2008 and later versions fail the DFS referral request when the value received in the **MaxReferralLevel** field of [SMB2_REQ_GET_DFS_REFERRAL \(section 2.2.2\)](#) is less than 3. Earlier versions of Windows return the referral formatted in the requested version.

<35> [Section 3.3.5.2:](#) Windows Server 2003 SP1 and later versions return a DFS referral version 3 response even if the DFS client specified DFS referral version 4.

<36> [Section 3.3.5.2:](#) Windows Server 2003 SP2 and later versions always retain the local domain information so that it doesn't get dropped off due to the server-side limit.

<37> [Section 3.3.5.2:](#) Windows servers have a server-side limit of 56 KB on the maximum size of a domain referral response. Within that limit, the server attempts to return all the [AD DS domain names](#) that it has. If the size of the domain referral entries is greater than 56 KB, it retains only 56 KB.

<38> [Section 3.3.5.2:](#) Windows returns a fixed value of 600 seconds.

<39> [Section 3.3.5.3:](#) Windows Server 2008 fails the DFS referral request when the value received in the **MaxReferralLevel** field of [SMB2_REQ_GET_DFS_REFERRAL \(section 2.2.2\)](#) is less than 3. Earlier versions of Windows return the referral formatted in the requested version.

<40> [Section 3.3.5.3:](#) Windows Server 2003 SP1 and later versions return a DFS referral version 3 response even if the DFS client specified DFS referral version 4.

<41> [Section 3.3.5.3:](#) Windows returns a fixed value of 600 seconds.

<42> [Section 3.3.5.4:](#) Windows Server 2008 and later versions fail the DFS referral request when the value received in the **MaxReferralLevel** field of [SMB2_REQ_GET_DFS_REFERRAL \(section 2.2.2\)](#) is less than 3. Earlier versions return the referral formatted in the requested version.

<43> [Section 3.3.5.4:](#) Windows 2000 Server does not support DFS referral site costing for sysvol referral responses. Windows Server 2003 and later versions, including Windows Server 2008, can be configured to support DFS referral site costing for sysvol referral responses. DFS referral site costing is disabled by default in Windows Server 2003, and enabled by default in Windows Server 2008. The registry value SiteCostedReferrals under the registry key HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Dfs\Parameters must be set to 1 at service startup to enable DFS referral site costing for sysvol referral responses in Windows Server 2003 versions.

<44> [Section 3.3.5.4:](#) Windows assigns infinite cost to a target if it cannot identify either the target AD DS site or the target AD DS site cost from the client AD DS site.

<45> [Section 3.3.5.4:](#) Windows randomizes the order of targets within each target set.

<46> [Section 3.3.5.4:](#) In Windows Server 2003 SP1 and later versions, the registry value SysvolNetlogonTargetFailback under the registry key HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Dfs\Parameters controls target failback for sysvol referrals.

[<47> Section 3.3.5.4:](#) Windows sets the **TimeToLive** field to a fixed value of 900 seconds.

7 Index

A

Abstract data model
 [client](#)
 [DC](#)
 [root target server](#)
AD DS domain name
[Applicability](#)
[Application-initiated I/O operation on UNC path](#)

C

[Capability negotiation](#)
Client
 [abstract data model](#)
 [higher-layer triggered events](#)
 [I/O operation on UNC path](#)
 I/O operations ([section 3.1.5.1](#), [section 3.1.5.2](#),
 [section 3.1.5.3](#))
 [initialization](#)
 [local events](#)
 [message processing](#)
 [overview](#)
 [referral response](#)
 requirements ([section 1.5.1](#), [section 1.5.2](#))
 [sequencing rules](#)
 [timer events](#)
 [timers](#)
[Conventions - syntax](#)

D

Data model - abstract
 [client](#)
 [DC](#)
 [root target server](#)
DC
 [abstract data model](#)
 [DC referral request](#)
 [DFS referral request](#)
 [domain referral request](#)
 [higher-layer triggered events](#)
 [initialization](#)
 [link referral request](#)
 [local events](#)
 [message processing](#)
 [overview](#)
 requirements ([section 1.5.1](#), [section 1.5.3](#))
 [root referral request](#)
 [sequencing rules](#)
 [sysvol referral request](#)
 [timer events](#)
 [timers](#)
[DC referral example](#)
DC referral request
 [DC](#)
 [root target server](#)
[DC referral response](#)
[DFS link](#)

[DFS link target](#)
DFS referral request
 [DC](#)
 [root target server](#)
DFS root
[DFS root target](#)
[DFS target](#)
[DFS_REFERRAL_V1 packet](#)
[DFS_REFERRAL_V2 packet](#)
[DFS_REFERRAL_V3](#)
[DFS_REFERRAL_V4](#)
[Domain referral example](#)
Domain referral request
 [DC](#)
 [root target server](#)
[Domain referral response](#)
[Domain-based DFS link referral example](#)
[Domain-based DFS root referral example](#)
[Domain-based DFS root referral packet trace example](#)
DVS_REFERRAL_V3 packet ([section 2.2.4.3.1](#), [section 2.2.4.3.2](#))

E

Examples
 [DC referral example](#)
 [domain referral example](#)
 [domain-based DFS link referral example](#)
 [domain-based DFS root referral example](#)
 [domain-based DFS root referral packet trace example](#)
 [overview](#)
 [stand-alone DFS root referral example](#)

F

[Fields - vendor-extensible](#)

G

[Glossary](#)

H

Higher-layer triggered events
 [client](#)
 [DC](#)
 [root target server](#)
[Hostname](#)

I

[I/O operation on UNC path](#)
I/O operations
 client ([section 3.1.5.1](#), [section 3.1.5.2](#), [section 3.1.5.3](#))
[Implementer - security considerations](#)
[Index of security parameters](#)
[Informative references](#)

Initialization

[client](#)
[DC](#)
[root target server](#)
[Introduction](#)

L

Link referral request
[DC](#)
[root target server](#)
[Link referral response](#)

Local events

[client](#)
[DC](#)
[root target server](#)

M

Message processing
[client](#)
[DC](#)
[root target server](#)

Messages

[overview](#)
[syntax](#)
[transport](#)

N

[Normative references](#)

O

[Overview \(synopsis\)](#)

P

[Parameters - security index](#)
[Preconditions](#)
[Prerequisites](#)

R

References
[informative](#)
[normative](#)
[overview](#)
[Referral entry types](#)
[Referral response - client](#)
[Relationship to other protocols](#)
[RESP_GET_DFS_REFERRAL packet](#)
Root referral request
[DC](#)
[root target server](#)
[Root referral response](#)
Root target server
[abstract data model](#)
[DC referral request](#)
[DFS referral request](#)
[domain referral request](#)

[higher-layer triggered events](#)
[initialization](#)
[link referral request](#)
[local events](#)
[message processing](#)
[overview](#)

requirements ([section 1.5.1](#), [section 1.5.3](#))
[root referral request](#)
[sequencing rules](#)
[sysvol referral request](#)
[timer events](#)
[timers](#)

S

Security

[implementer considerations](#)
[overview](#)
[parameter index](#)

Sequencing rules

[client](#)
[DC](#)
[root target server](#)

Sharename

[SMB2_REQ_GET_DFS_REFERRAL packet](#)
[Stand-alone DFS root referral example](#)
[Standards assignments](#)

STATUS_PATH_NOT_COVERED ([section 3.1.5.1](#),
[section 3.1.5.2](#))

Syntax

[conventions](#)
[overview](#)
[referral entry types](#)

sysvol referral request

[DC](#)
[root target server](#)
[sysvol referral response](#)

T

Timer events

[client](#)
[DC](#)
[root target server](#)

Timers

[client](#)
[DC](#)
[root target server](#)

Transport

Triggered events - higher-layer

[client](#)
[DC](#)
[root target server](#)

U

[UNC path](#)
[UNC path - I/O operation](#)
[User-initiated I/O operation on UNC path](#)

V

[Vendor-extensible fields](#)
[Versioning](#)

W

[Windows behavior](#)