

# [MS-DPSP]: Digest Protocol Extensions

---

## Intellectual Property Rights Notice for Open Specifications Documentation

- **Technical Documentation.** Microsoft publishes Open Specifications documentation for protocols, file formats, languages, standards as well as overviews of the interaction among each of these technologies.
- **Copyrights.** This documentation is covered by Microsoft copyrights. Regardless of any other terms that are contained in the terms of use for the Microsoft website that hosts this documentation, you may make copies of it in order to develop implementations of the technologies described in the Open Specifications and may distribute portions of it in your implementations using these technologies or your documentation as necessary to properly document the implementation. You may also distribute in your implementation, with or without modification, any schema, IDL's, or code samples that are included in the documentation. This permission also applies to any documents that are referenced in the Open Specifications.
- **No Trade Secrets.** Microsoft does not claim any trade secret rights in this documentation.
- **Patents.** Microsoft has patents that may cover your implementations of the technologies described in the Open Specifications. Neither this notice nor Microsoft's delivery of the documentation grants any licenses under those or any other Microsoft patents. However, a given Open Specification may be covered by Microsoft's Open Specification Promise (available here: <http://www.microsoft.com/interop/osp>) or the Community Promise (available here: <http://www.microsoft.com/interop/cp/default.aspx>). If you would prefer a written license, or if the technologies described in the Open Specifications are not covered by the Open Specifications Promise or Community Promise, as applicable, patent licenses are available by contacting [iplg@microsoft.com](mailto:iplg@microsoft.com).
- **Trademarks.** 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.
- **Fictitious Names.** The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted in this documentation are fictitious. No association with any real company, organization, product, domain name, email address, logo, person, place, or event is intended or should be inferred.

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

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

## Revision Summary

Date	Revision History	Revision Class	Comments
03/14/2007	1.0	Major	Updated and revised the technical content.
04/10/2007	1.1	Minor	Updated the technical content.
05/18/2007	2.0	Major	New format; Updated technical content
06/08/2007	2.0.1	Editorial	Revised and edited the technical content.
07/10/2007	2.0.2	Editorial	Revised and edited the technical content.
08/17/2007	2.0.3	Editorial	Revised and edited the technical content.
09/21/2007	2.0.4	Editorial	Revised and edited the technical content.
10/26/2007	2.0.5	Editorial	Revised and edited the technical content.
01/25/2008	2.0.6	Editorial	Revised and edited the technical content.
03/14/2008	2.0.7	Editorial	Revised and edited the technical content.
06/20/2008	2.0.8	Editorial	Revised and edited the technical content.
07/25/2008	2.0.9	Editorial	Revised and edited the technical content.
08/29/2008	2.1	Minor	Remove incorrect behavior note in section 1.3.1 and update informative information in section 1.3.1.
10/24/2008	2.1.1	Editorial	Revised and edited the technical content.
12/05/2008	3.0	Major	Updated and revised the technical content.
01/16/2009	3.0.1	Editorial	Revised and edited the technical content.
02/27/2009	3.0.2	Editorial	Revised and edited the technical content.
04/10/2009	3.0.3	Editorial	Revised and edited the technical content.
05/22/2009	4.0	Major	Updated and revised the technical content.
07/02/2009	5.0	Major	Updated and revised the technical content.
08/14/2009	6.0	Major	Updated and revised the technical content.
09/25/2009	6.1	Minor	Updated the technical content.
11/06/2009	6.1.1	Editorial	Revised and edited the technical content.
12/18/2009	6.1.2	Editorial	Revised and edited the technical content.
01/29/2010	7.0	Major	Updated and revised the technical content.
03/12/2010	7.0.1	Editorial	Revised and edited the technical content.

<b>Date</b>	<b>Revision History</b>	<b>Revision Class</b>	<b>Comments</b>
04/23/2010	7.0.2	Editorial	Revised and edited the technical content.
06/04/2010	7.0.3	Editorial	Revised and edited the technical content.
07/16/2010	7.0.3	No change	No changes to the meaning, language, or formatting of the technical content.
08/27/2010	7.0.3	No change	No changes to the meaning, language, or formatting of the technical content.
10/08/2010	7.0.3	No change	No changes to the meaning, language, or formatting of the technical content.
11/19/2010	7.0.3	No change	No changes to the meaning, language, or formatting of the technical content.
01/07/2011	7.0.3	No change	No changes to the meaning, language, or formatting of the technical content.
02/11/2011	7.0.3	No change	No changes to the meaning, language, or formatting of the technical content.

# Contents

<b>1 Introduction</b>	<b>6</b>
1.1 Glossary	6
1.2 References	6
1.2.1 Normative References	6
1.2.2 Informative References	7
1.3 Overview	7
1.4 Relationship to Other Protocols	8
1.5 Prerequisites/Preconditions	8
1.6 Applicability Statement	8
1.7 Versioning and Capability Negotiation	8
1.8 Vendor-Extensible Fields	8
1.9 Standards Assignments	8
<b>2 Messages</b>	<b>9</b>
2.1 Transport	9
2.2 Message Syntax	9
<b>3 Protocol Details</b>	<b>10</b>
3.1 Common Details	10
3.1.1 Abstract Data Model	10
3.1.2 Timers	10
3.1.3 Initialization	10
3.1.4 Higher-Layer Trigger Events	10
3.1.5 Processing Events and Sequencing Rules	10
3.1.5.1 Authentication-Info	10
3.1.5.2 Realm Directive	10
3.1.5.3 Subsequent Authentication	10
3.1.5.4 Opaque Directive	10
3.1.6 Timer Events	11
3.1.7 Other Local Events	11
3.2 Client Details	11
3.2.1 Abstract Data Model	11
3.2.2 Timers	11
3.2.3 Initialization	11
3.2.4 Higher-Layer Trigger Events	11
3.2.5 Processing Events and Sequencing Rules	11
3.2.5.1 Client Nonce Generation	11
3.2.5.2 Qop Directive	11
3.2.6 Timer Events	12
3.2.7 Other Local Events	12
3.3 Server Details	12
3.3.1 Abstract Data Model	12
3.3.2 Timers	12
3.3.3 Initialization	12
3.3.4 Higher-Layer Trigger Events	12
3.3.5 Processing Events and Sequencing Rules	12
3.3.5.1 Server Nonce Generation	12
3.3.5.2 Qop-Options Directive	12
3.3.5.3 Realm Directive for the Digest Challenge	13
3.3.5.4 Principal Name Validation	13

3.3.5.5 Host Name.....	13
3.3.6 Timer Events .....	13
3.3.7 Other Local Events .....	13
<b>4 Protocol Examples .....</b>	<b>14</b>
<b>5 Security .....</b>	<b>16</b>
5.1 Security Considerations for Implementers .....	16
5.2 Index of Security Parameters .....	16
<b>6 Appendix A: Product Behavior .....</b>	<b>17</b>
<b>7 Change Tracking.....</b>	<b>20</b>
<b>8 Index .....</b>	<b>21</b>

# 1 Introduction

The Microsoft Windows® implementation of the Digest Authentication Protocol contains variations from the Digest Authentication standard specified in [\[RFC2617\]](#) and [\[RFC2831\]](#).

Digest authentication supports client authentication to servers (based on the user's name and password) and server authentication to the client.

Windows implements the digest access authentication for HTTP/1.1 as specified in [\[RFC2617\]](#). Windows also implements digest authentication as a Simple Authentication and Security Layer (SASL) mechanism, as specified in [\[RFC2831\]](#).

Higher-Layer protocols such as Lightweight Directory Access Protocol (LDAP) ([\[RFC2251\]](#)) employ digest authentication as an SASL mechanism. The Windows implementation is compliant with digest authentication, as specified in [\[RFC2617\]](#) and [\[RFC2831\]](#).

This protocol is also how Windows implements optional fields and behaviors (specified by keywords such as MAY or SHOULD) and how Windows implements support for older clients and servers that exhibit nonconforming behavior to [\[RFC2617\]](#) and [\[RFC2831\]](#).

## 1.1 Glossary

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

**domain name (2)**  
**keyed hash**  
**nonce**

The following terms are specific to this document:

**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](mailto: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.

[FIPS140] National Institute of Standards and Technology, "Federal Information Processing Standards Publication 140-2: Security Requirements for Cryptographic Modules", December 2002, <http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf>

[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>

[RFC2222] Myers, J., "Simple Authentication and Security Layer (SASL)", RFC 2222, October 1997, <http://www.ietf.org/rfc/rfc2222.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>

[RFC2616] Fielding, R., Gettys, J., Mogul, J., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, <http://www.ietf.org/rfc/rfc2616.txt>

[RFC2617] Franks, J., Hallam-Baker, P., Hostetler, J., et al., "HTTP Authentication: Basic and Digest Access Authentication", RFC 2617, June 1999, <http://www.ietf.org/rfc/rfc2617.txt>

[RFC2829] Wahl, M., Alvestrand, H., Hodges, J., and Morgan, R., "Authentication Methods for LDAP", RFC 2829, May 2000, <http://www.ietf.org/rfc/rfc2829.txt>

[RFC2831] Leach, P., and Newman, C., "Using Digest Authentication as a SASL Mechanism", RFC 2831, May 2000, <http://www.ietf.org/rfc/rfc2831.txt>

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

## 1.2.2 Informative References

[DRAFT-DIGESTBIND] Kivinen, T., Huttunen, A., Swander, B., and Volpe, V., "Channel binding for HTTP Digest Authentication", July 2008, <http://tools.ietf.org/html/draft-santesson-digestbind-01>

[MS-APDS] Microsoft Corporation, "[Authentication Protocol Domain Support Specification](#)", July 2006.

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

[MS-KILE] Microsoft Corporation, "[Kerberos Protocol Extensions](#)", July 2006.

[RFC2069] Franks, J., Hallam-Baker, P., Hostetler, J., et al., "An Extension to HTTP: Digest Access Authentication", RFC 2069, January 1997, <http://www.ietf.org/rfc/rfc2069.txt>

## 1.3 Overview

The digest authentication mechanism [\[RFC2617\]](#) [\[RFC2831\]](#) performs authentication between a client and a server based on a user name and a password. The digest authentication protocol can authenticate the client to the server, and optionally the server to a client; the latter is termed mutual authentication.

In the digest authentication mechanism, the client is presented with a **nonce**, a randomly generated value sent by the server to the client. The client proves knowledge of the password by computing a **keyed hash** over parameters sent by the server and parameters generated locally by the client.

Once the server has validated the keyed hash by performing the same computation, it can authenticate itself to the client by computing another keyed hash and returning it to the client. Because the correct keyed hash results can only be created by someone who knows the password, the two parties are assured of the other knowing the password.

Subsequent client-to-server messages can be authenticated by a keyed hash. Replay protection is provided via an ordered nonce count, as specified in [\[RFC2831\]](#). Digest authentication as an SASL mechanism expands the digest access authentication protocol by also supporting integrity and confidentiality of messages sent between the client and server. For more information on SASL, see [\[RFC2831\]](#). For more information on digest access authentication, see [\[RFC2617\]](#).

## 1.4 Relationship to Other Protocols

The Digest Authentication Protocol was originally specified as a native authentication method for HTTP/1.1, as specified in [\[RFC2616\]](#), to serve as an improvement on the HTTP basic authentication. The popularity of digest authentication grew, and it was covered as an SASL [\[RFC2222\]](#) mechanism by the specification in [\[RFC2831\]](#). Once made into an SASL mechanism, the Digest Authentication Protocol became available for other protocols, such as Lightweight Directory Access Protocol (LDAP), as specified in [\[RFC2251\]](#).<1>

## 1.5 Prerequisites/Preconditions

Digest Access Authentication: Microsoft Extensions assumes the following:

1. Prior to the start of digest authentication, the client and the server have access to the user's password (shared knowledge between them).
2. On the server and the client, a source of cryptographically useful random numbers is available for generating a nonce.<2>

## 1.6 Applicability Statement

The digest authentication mechanism is used in environments that require users to authenticate to servers to access secure resources. Note that Kerberos (for more information, see [\[MS-KILE\]](#)) and public key-based authentication offer stronger security guarantees both in terms of initial authentication and subsequent confidentiality and integrity of client/server traffic.

The digest authentication mechanism can be used in environments where these stronger mechanisms are not available and can serve for interoperability purposes with multiple vendors, browsers, Web servers, and directory services.<3>

## 1.7 Versioning and Capability Negotiation

Neither the Digest Authentication Protocol nor the Digest Access Authentication: Microsoft Extensions have any versioning capability. The Digest Authentication Protocol does have support for negotiating what cryptographic algorithms to use. This is specified in [\[RFC2831\]](#) section 2.1.2.

## 1.8 Vendor-Extensible Fields

The Digest Access Authentication: Microsoft Extensions introduce no additional vendor-extensible fields beyond those specified in [\[RFC2831\]](#) and [\[RFC2617\]](#).<4>

## 1.9 Standards Assignments

There are no standards assignments in the Digest Access Authentication: Microsoft Extensions beyond what is specified in [\[RFC2617\]](#) and [\[RFC2831\]](#).



## 2 Messages

### 2.1 Transport

The elements of the Digest Access Authentication: Microsoft Extensions messages are embedded directly in HTTP/1.1 messages [\[RFC2616\]](#) when digest authentication is used within HTTP, as specified in [\[RFC2617\]](#). As a result, Digest Access Authentication: Microsoft Extensions uses the HTTP/1.1 transport, as specified in [\[RFC2616\]](#).

As specified in [\[RFC2831\]](#), Digest Access Authentication: Microsoft Extensions messages are carried via SASL [\[RFC2222\]](#) in SASL-aware protocols, such as LDAP.

An extension to add channel binding to the HTTP Digest Authentication protocol has been submitted as a draft standard to the IETF. The client and server MAY [<5>](#) support channel binding as specified in [\[DRAFT-DIGESTBIND\]](#).

### 2.2 Message Syntax

The message syntax of the Digest Access Authentication: Microsoft Extensions is as specified in [\[RFC2617\]](#) and [\[RFC2831\]](#).[<6>](#)

When processing the **username** field, the server SHOULD process "\" characters as improperly escaped "\\\" characters.

The extension to this protocol consists of the use of the capability specified in [\[RFC2617\]](#) to add a new directive via the auth-param in the digest-challenge message ([\[RFC2831\]](#) section 2.1.1). The server sends: charset=utf-8 in the digest-challenge message. This indicates that the server can process UTF-8 encoded strings and that the client might use [\[UNICODE\]](#) encoding for the **username** field and in the password if it can also process UTF-8. Clients SHOULD use [\[UNICODE\]](#) encoding when it is offered by the server to allow authentication with a region's supported character sets.[<7>](#)

## 3 Protocol Details

The following sections specify details of the Digest Access Authentication: Microsoft Extensions, including abstract data models and message processing rules that are common for both the client and the server. The variations are as specified in [\[RFC2617\]](#) and [\[RFC2831\]](#).

### 3.1 Common Details

#### 3.1.1 Abstract Data Model

The abstract data model follows what is specified in [\[RFC2617\]](#) and [\[RFC2831\]](#). In addition, there is the following variable:

**Integrity:** When this value is set, it indicates that the higher-layer protocol requires integrity in addition to simple authentication. This corresponds to the auth-int option, as specified in [\[RFC2617\]](#) sections 3.2.1 and 3.2.2. If this is not set, only auth is specified as the requested quality of protection.

#### 3.1.2 Timers

There are no common timers specified in the Digest Access Authentication: Microsoft Extensions.

#### 3.1.3 Initialization

The random-number generator is initialized for nonce and key generation.<8>

#### 3.1.4 Higher-Layer Trigger Events

Digest Access Authentication: Microsoft Extensions are triggered by a higher-layer application protocol, such as when HTTP or LDAP creates a connection and requires authentication. The higher-layer protocol determines if the **Integrity** option is enabled for a particular connection.<9>

#### 3.1.5 Processing Events and Sequencing Rules

##### 3.1.5.1 Authentication-Info

Specifying fields in the Authentication-Info message sent on the third leg of the Digest Access Authentication: Microsoft Extensions from the server to the client SHOULD be supported, as specified in [\[RFC2617\]](#) section 3.2.3.<10>

##### 3.1.5.2 Realm Directive

The realm directive is optional; if not present, the client SHOULD solicit it from the user or be able to compute a default, as specified in [\[RFC2831\]](#) section 2.1.1.<11>

##### 3.1.5.3 Subsequent Authentication

Digest Protocol Extensions does not support "subsequent authentication" ([\[RFC2831\]](#) section 2.2) when used as a SASL mechanism.

##### 3.1.5.4 Opaque Directive

The opaque directive is a string of data that SHOULD be returned by the client unchanged in the authorization header, as specified in [\[RFC2617\]](#) section 3.2.1.<12>

### 3.1.6 Timer Events

There are no common timer events specified in the Digest Access Authentication: Microsoft Extensions.

### 3.1.7 Other Local Events

There are no additional local events specified in the Digest Access Authentication: Microsoft Extensions.

## 3.2 Client Details

### 3.2.1 Abstract Data Model

The client must maintain a nonce for ongoing communications with the server, as specified in [\[RFC2617\]](#) section 3.2.2 and [\[RFC2831\]](#) section 2.1.2. The client maintains the state of the nonce by keeping track of the nonce count, which is incremented each time the client sends a message to the server. <13>

**ClientCompat\_QuotedQop**: A Boolean value indicating whether the client will send quoted qop directive values. [\[RFC2617\]](#) 3.2.2 specifies that the client qop directive value should be unquoted. When **ClientCompat\_QuotedQop** is TRUE, the client will send qop values as quoted directive values. When **ClientCompat\_QuotedQop** is FALSE, the client will send qop values as unquoted directive values. This value SHOULD be initialized to TRUE.

### 3.2.2 Timers

There are no additional client-specific timers specified in the Digest Access Authentication: Microsoft Extensions.

### 3.2.3 Initialization

For details on initialization for the client, see section [3.1.3](#).

### 3.2.4 Higher-Layer Trigger Events

For details on higher-layer trigger events for the client, see section [3.1.4](#).

### 3.2.5 Processing Events and Sequencing Rules

#### 3.2.5.1 Client Nonce Generation

It is highly recommended that when the **nonce** is generated by the client, it contains at least 64 bits of entropy, as specified in [\[RFC2831\]](#) section 2.1.2. <14>

#### 3.2.5.2 Qop Directive

The qop directive, as specified in [\[RFC2617\]](#) section 3.2.2, is optional to preserve backward compatibility with minimal implementation of digest access authentication, as specified in [\[RFC2069\]](#). [\[RFC2617\]](#) specifies that the qop directive SHOULD be used by the client (if the server indicates that qop is supported) by providing a qop directive in the WWW-Authenticate header field. <15>

The server MUST treat single unquoted qop values, such as 'qop-value', the same as quoted qop values, such as 'qop="value"'.<16>

### 3.2.6 Timer Events

There are no additional client-specific timer events specified in the Digest Access Authentication: Microsoft Extensions.

### 3.2.7 Other Local Events

There are no additional client-specific local events specified in the Digest Access Authentication: Microsoft Extensions.

## 3.3 Server Details

### 3.3.1 Abstract Data Model

The server must associate state with each authenticated connection, as specified in [\[RFC2617\]](#). Specifically, the server must associate a client and a server nonce with each connection, along with a nonce count ([\[RFC2617\]](#) section 3.2.2 and [\[RFC2831\]](#) section 2.1.2) for ongoing communications. The server may keep all this information longer than an active connection, depending on the length of time that is allotted for subsequent authentication, as specified in [\[RFC2617\]](#) section 3.3.

### 3.3.2 Timers

There are no additional server-specific timers specified in the Digest Access Authentication: Microsoft Extensions.

### 3.3.3 Initialization

For details on initialization for the server, see section [3.1.3](#).

### 3.3.4 Higher-Layer Trigger Events

For details on the higher-layer trigger events for the server, see section [3.1.4](#).

### 3.3.5 Processing Events and Sequencing Rules

#### 3.3.5.1 Server Nonce Generation

It is recommended that the nonce computed by the server contain at least 64 bits of entropy, as specified in [\[RFC2831\]](#) section 2.1.1.<17>

#### 3.3.5.2 Qop-Options Directive

The qop-options directive, as specified in [\[RFC2617\]](#) section 3.2.1, is optional; but it is used for backward compatibility with digest access authentication, as specified in [\[RFC2069\]](#). The qop-options directive SHOULD be used by all implementations compliant with this version of the digest authentication mechanism and SHOULD be enclosed in quotation marks.

### 3.3.5.3 Realm Directive for the Digest Challenge

The [realm directive](#) is required if the server provides any realms in the digest challenge; in which case it may appear exactly once, and its value SHOULD be one of those realms, as specified in [\[RFC2831\]](#) section 2.1.2.

### 3.3.5.4 Principal Name Validation

Digest-Uri indicates the principal name of the service that the client is attempting to connect with, as specified in [\[RFC2831\]](#) section 2.1.2. Servers SHOULD check that the supplied value is correct. [<18>](#)

### 3.3.5.5 Host Name

The Digest Access Authentication: Microsoft Extensions do not make use of the host field.

### 3.3.6 Timer Events

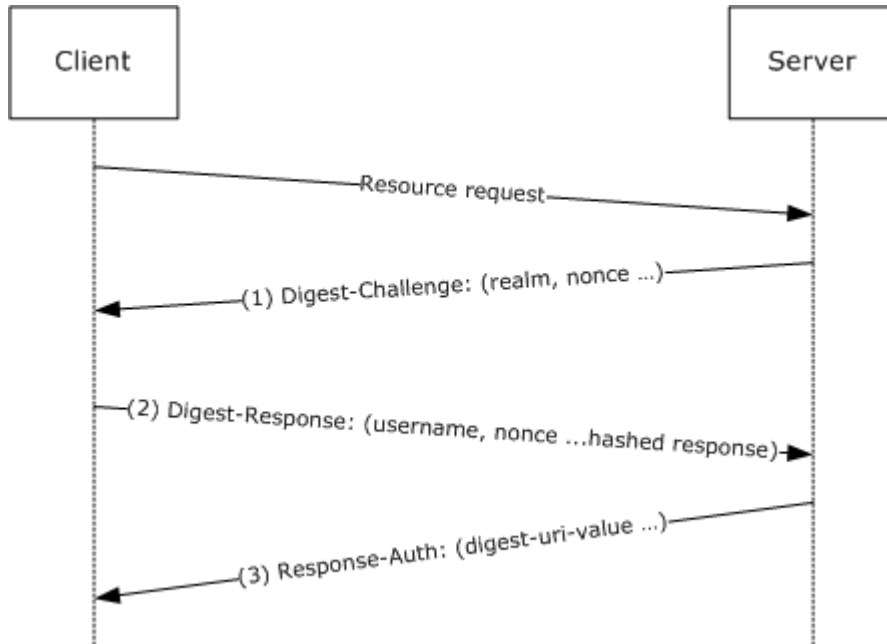
There are no additional server-specific timer events specified in the Digest Access Authentication: Microsoft Extensions.

### 3.3.7 Other Local Events

There are no additional server-specific local events specified in the Digest Access Authentication: Microsoft Extensions.

## 4 Protocol Examples

The following diagram and procedural steps describe a common scenario to illustrate the function of the Digest Access Authentication: Microsoft Extensions.



**Figure 1: Common use scenario for Digest Access Authentication: Microsoft Extensions**

1. After the client attempts to access a protected resource (for example, ResourceA) on the server, the server returns a digest-challenge message to the client. Among other fields, the digest-challenge message includes a randomly generated nonce and the **domain name** of the server (via the **realm** field). A sample digest Challenge is shown here.

```
qop="auth",algorithm=MD5-sess,
nonce="91c121b4f47ec601a281ceefaa5d6f2b096897ea0797fdd2ea72bfeec7fd
a64433a98d4ae57186a1",
charset=utf-8,realm="TestDomain"
```

2. The client obtains the user name (for example, User123) and password for the user and constructs a response to the server's challenge. In the digest-response, the client proves knowledge of the user's password by performing a keyed hash over the user name, nonce, and other fields (the password is fed into the hash). A sample digest-response is shown here.

```
username="User123",
realm="TestDomain",
qop="auth",
algorithm="MD5-sess",
uri="/ResourceA",
nonce="91c121b4f47ec601a281ceefaa5d6f2b096897ea0797fdd2ea72bfeec7fd
a64433a98d4ae57186a1",
nc=00000001,
cnonce="579c5e7723ad0ef8eeb0b7427379bdd4",
```

response="641b92d2d8af170329ce308832a4df13"

3. The server validates the digest-response message by looking up the user's password by using the user name that the client sent, recomputing the keyed hash over fields from the digest-response message, and then comparing the resulting hash value to the Response directive value sent by the client. If the values match, the client's digest-response message is valid; otherwise, the authentication request fails. The server further checks that the client sent the expected nonce and nonce-count values (and not an old, replayed value). After the digest response is validated, the client is authenticated to the server. For mutual authentication, the server has the option to send a keyed hash over the URI that the client requested and return it to the client in the Response-Auth message. Note that sending the Response-Auth message only applies to digest authentication when used as an SASL mechanism, as specified in [\[RFC2831\]](#). For further information, see section [3.1.5.1](#).

## 5 Security

### 5.1 Security Considerations for Implementers

Kerberos (for more information, see [\[MS-KILE\]](#)) and public key-based authentication offer stronger security guarantees both in terms of initial authentication and in subsequent confidentiality and integrity of client-server traffic. The digest authentication mechanism can be used in environments where these stronger mechanisms are not available.

### 5.2 Index of Security Parameters

There are no security parameters specified in the Digest Access Authentication: Microsoft Extensions.



## 6 Appendix A: Product Behavior

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

- Microsoft Windows NT® operating system
- Microsoft Windows® 2000 operating system
- Windows® XP operating system
- Windows Server® 2003 operating system
- Windows Vista® operating system
- Windows Server® 2008 operating system
- Windows® 7 operating system
- Windows Server® 2008 R2 operating system

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

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

[<1> Section 1.4:](#) In Windows domain environments, validating the digest authentication exchange can be performed at a different computer (for more information, see [\[MS-APDS\]](#)). This does not affect the actual implementation of the digest authentication between the client and the server. For more information on digest authentication validation, see [\[RFC2617\]](#) section 3.2.2. Windows NT does not have support for [\[MS-DPSP\]](#).

[<2> Section 1.5:](#) The random number generator used in Windows is FIPS-140-compliant. For information on the FIPS-140 random number generator requirements, see [\[FIPS140\]](#) sections 4.7.1, 4.9.1, and 4.9.2.

[<3> Section 1.6:](#) In Windows environments, the Digest Authentication Protocol can be used for authentication during HTTP traffic on Windows 2000, Windows Server 2003, Windows Server 2008, and Windows Server 2008 R2 through the use and configuration of Internet Information Server (IIS). In addition, LDAP clients and servers in an Active Directory domain can make use of the SASL mechanism for digest authentication.

The LDAP server in Active Directory offers digest authentication by default as one of several authentication mechanisms for Active Directory. This is exposed by the supported SASL Mechanism attribute, as specified in [\[RFC2829\]](#). All native use of LDAP within Windows uses SPNEGO authentication, and not digest authentication.

[<4> Section 1.8:](#) Windows uses the capability specified in [\[RFC2617\]](#) to add a new directive via the auth-param in the digest-challenge message ([\[RFC2831\]](#) section 2.1.1). The server sends: charset=utf-8 in the **auth-param** field. This indicates that the server can process UTF-8 encoded strings and that the client might use Unicode encoding for the **username** field and in the password

if it can also process UTF-8. Windows clients will use [\[UNICODE\]](#) encoding when it is offered by the server to allow authentication with a region's supported character sets.

[<5> Section 2.1:](#) In Windows 7 and Windows Server 2008 R2, the Digest protocol was further extended to support channel binding for HTTP Digest Authentication [\[DRAFT-DIGESTBIND\]](#).

[<6> Section 2.2:](#) When acting as a server, Windows XP, Windows Server 2003, Windows Server 2008, and Windows Server 2008 R2 attempt to decode the **username** field of the digest-response message ([\[RFC2617\]](#) section 3.2.2) using the RFC-compliant character escaping ([\[RFC2616\]](#) section 2.2).

However, if no account with that name can be found, or if the keyed hash computed by the server does not match the keyed hash sent by the client, the server attempts to decode the original username by using backslash '\' characters that are not properly escaped (treated as if the client specified '\\'), and then retries. Windows NT and Windows XP incorrectly escape the backslash '\' character in the account name. If a backslash ('\') character is presented by the client, the string is treated as "NetbiosDomainName\\AccountName".

Unlike other authentication protocols, digest authentication does not support the diacritical folding that is applied by Active Directory.

[<7> Section 2.2:](#) The auth param extensions are not supported on Windows NT and Windows 2000.

[<8> Section 3.1.3:](#) The random number generator for keys and nonces is initialized by other components, but complies with what is specified in [\[FIPS140\]](#).

[<9> Section 3.1.4:](#) For details on where exactly digest authentication is used in the Windows environment, see section [1.6](#).

[<10> Section 3.1.5.1:](#) The Windows 2000, Windows XP, Windows Server 2003, Windows Server 2008, and Windows Server 2008 R2 digest implementations for HTTP do not support the [Authentication-Info](#) message. If a third-party server generates an [Authentication-Info](#) message, it will be ignored on the Windows client. However, the third leg of the Digest Access Authentication: Microsoft Extensions as the SASL mechanism is supported, as specified in [\[RFC2831\]](#) section 2.1.3.

[<11> Section 3.1.5.2:](#) The [realm directive](#) is used in Windows 2000, Windows XP, Windows Server 2003, Windows Server 2008, and Windows Server 2008 R2. The realm is set to the domain name of the server by default. This can be overridden at the server by configuration options. The user is prompted and has the chance to override the realm value sent by the server (that is, the user can enter a realm other than the one sent by the server).

[<12> Section 3.1.5.4:](#) For Windows 2000, the opaque value is returned from the client to the server only during the initial authentication step. They do not return the opaque value for subsequent authentication ([\[RFC2617\]](#) section 3.3).

[<13> Section 3.2.1:](#) The Windows digest implementation ensures that if the nonce count value wraps, the authentication fails.

[<14> Section 3.2.5.1:](#) For Windows XP, Windows Vista, Windows Server 2003, Windows 7, Windows Server 2008, and Windows Server 2008 R2, the **cnonce** contains 128 bits of entropy by using a random-number generator, as specified in [\[FIPS140\]](#).

[<15> Section 3.2.5.2:](#) Windows 2000, Windows XP, Windows Server 2003, Windows Server 2008, and Windows Server 2008 R2: Windows servers indicate support for different qops, and clients select accordingly, using the qop values as specified in [\[RFC2617\]](#) section 3.2.1 (subject to exceptions specified in section [2.2](#)) and in [\[RFC2831\]](#) section 2.1.1. The value of this directive

depends on the quality of protection requested by the calling application. For information on Windows usage of the [qop directive](#), see section [2.2](#).

[<16> Section 3.2.5.2](#): Windows 2000 Server accepts only quoted qop directive values as with 'qop="value"'.

[<17> Section 3.3.5.1](#): In Windows Server 2003, Windows Server 2008, and Windows Server 2008 R2, the nonce contains 128 bits of entropy by using a random-number generator as specified in [\[FIPS140\]](#).

[<18> Section 3.3.5.4](#): The digest implementation does not do any validation of the digest-uri value. The application that calls the digest authentication validates the principal name specified by the digest-uri. The digest authentication implementation returns the principal name to the calling application.

## 7 Change Tracking

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

## 8 Index

### A

Abstract data model  
  client ([section 3.1.1](#) 10, [section 3.2.1](#) 11)  
  server ([section 3.1.1](#) 10, [section 3.3.1](#) 12)  
[Applicability statement](#) 8  
[Authentication](#) 10  
[AuthenticationInfo](#) 10

### C

[Capability negotiation](#) 8  
[Change tracking](#) 20  
Client  
  abstract data model ([section 3.1.1](#) 10, [section 3.2.1](#) 11)  
  higher-layer trigger events ([section 3.1.4](#) 10, [section 3.2.4](#) 11)  
  initialization ([section 3.1.3](#) 10, [section 3.2.3](#) 11)  
  local events ([section 3.1.7](#) 11, [section 3.2.7](#) 12)  
  message processing ([section 3.1.5](#) 10, [section 3.2.5](#) 11)  
  message sequencing ([section 3.1.5](#) 10, [section 3.2.5](#) 11)  
  [opaque directive](#) 10  
  timer events ([section 3.1.6](#) 11, [section 3.2.6](#) 12)  
  timers ([section 3.1.2](#) 10, [section 3.2.2](#) 11)

### D

Data model – abstract  
  client ([section 3.1.1](#) 10, [section 3.2.1](#) 11)  
  server ([section 3.1.1](#) 10, [section 3.3.1](#) 12)

### E

[Examples](#) 14

### F

[Fields – vendor-extensible](#) 8

### G

[Glossary](#) 6

### H

Higher-layer trigger events  
  client ([section 3.1.4](#) 10, [section 3.2.4](#) 11)  
  server ([section 3.1.4](#) 10, [section 3.3.4](#) 12)  
[Host name](#) 13

### I

[Implementers – security considerations](#) 16  
[Informative references](#) 7  
Initialization  
  client ([section 3.1.3](#) 10, [section 3.2.3](#) 11)

  server ([section 3.1.3](#) 10, [section 3.3.3](#) 12)  
[Introduction](#) 6

### L

Local events  
  client ([section 3.1.7](#) 11, [section 3.2.7](#) 12)  
  server ([section 3.1.7](#) 11, [section 3.3.7](#) 13)

### M

Message processing  
  client ([section 3.1.5](#) 10, [section 3.2.5](#) 11)  
  server ([section 3.1.5](#) 10, [section 3.3.5](#) 12)  
Message sequencing  
  client ([section 3.1.5](#) 10, [section 3.2.5](#) 11)  
  server ([section 3.1.5](#) 10, [section 3.3.5](#) 12)  
Messages  
  syntax ([section 2](#) 9, [section 2.2](#) 9)  
  transport ([section 2](#) 9, [section 2.1](#) 9)

### N

Name  
  [host](#) 13  
  [validation - principal](#) 13  
Nonce  
  [client](#) 11  
  [server](#) 12  
[Normative references](#) 6

### O

Opaque directive  
  [client](#) 10  
  [server](#) 10  
[Overview](#) 7

### P

[Parameters - security](#) 16  
[Preconditions](#) 8  
[Prerequisites](#) 8  
[Principal name validation](#) 13  
[Product behavior](#) 17

### Q

[Qop directive](#) 11  
[Qop-options directive](#) 12

### R

Realm directive ([section 3.1.5.2](#) 10, [section 3.3.5.3](#) 13)  
References  
  [informative](#) 7  
  [normative](#) 6  
[Relationship to other protocols](#) 8

## S

[Security](#) 16

Sequencing – message

client ([section 3.1.5](#) 10, [section 3.2.5](#) 11)

server ([section 3.1.5](#) 10, [section 3.3.5](#) 12)

Server

abstract data model ([section 3.1.1](#) 10, [section 3.3.1](#) 12)

higher-layer trigger events ([section 3.1.4](#) 10, [section 3.3.4](#) 12)

initialization ([section 3.1.3](#) 10, [section 3.3.3](#) 12)

local events ([section 3.1.7](#) 11, [section 3.3.7](#) 13)

message processing ([section 3.1.5](#) 10, [section 3.3.5](#) 12)

message sequencing ([section 3.1.5](#) 10, [section 3.3.5](#) 12)

[opaque directive](#) 10

timer events ([section 3.1.6](#) 11, [section 3.3.6](#) 13)

timers ([section 3.1.2](#) 10, [section 3.3.2](#) 12)

[Standards assignments](#) 8

Syntax - message ([section 2](#) 9, [section 2.2](#) 9)

## T

Timer events

client ([section 3.1.6](#) 11, [section 3.2.6](#) 12)

server ([section 3.1.6](#) 11, [section 3.3.6](#) 13)

Timers

client ([section 3.1.2](#) 10, [section 3.2.2](#) 11)

server ([section 3.1.2](#) 10, [section 3.3.2](#) 12)

[Tracking changes](#) 20

Transport – message ([section 2](#) 9, [section 2.1](#) 9)

Trigger events – higher layer

client ([section 3.1.4](#) 10, [section 3.2.4](#) 11)

server ([section 3.1.4](#) 10, [section 3.3.4](#) 12)

## V

[Validation - principal name](#) 13

[Vendor-extensible fields](#) 8

[Versioning](#) 8