

[MS-OFBA]: Office Forms Based Authentication Protocol Specification

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.
- **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
05/16/2008	0.1		Initial Availability
10/06/2008	0.2	Editorial	Revised and edited the technical content
01/16/2009	1.0	Major	Revised and edited the technical content
07/13/2009	1.01	Major	Changes made for template compliance
08/28/2009	1.02	Editorial	Revised and edited the technical content
11/06/2009	1.03	Editorial	Revised and edited the technical content
02/19/2010	2.0	Editorial	Revised and edited the technical content
03/31/2010	2.01	Editorial	Revised and edited the technical content
04/30/2010	2.02	Editorial	Revised and edited the technical content
06/07/2010	2.03	Editorial	Revised and edited the technical content
06/29/2010	2.04	Minor	Clarified the meaning of the technical content.
07/23/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
09/27/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
11/15/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.
12/17/2010	2.04	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction	5
1.1 Glossary	5
1.2 References	5
1.2.1 Normative References	5
1.2.2 Informative References	5
1.3 Protocol Overview (Synopsis)	6
1.4 Relationship to Other Protocols	7
1.5 Prerequisites/Preconditions	7
1.6 Applicability Statement	7
1.7 Versioning and Capability Negotiation	7
1.8 Vendor-Extensible Fields	7
1.9 Standards Assignments	7
2 Messages	8
2.1 Transport	8
2.2 Message Syntax	8
2.2.1 Protocol Discovery Requests	8
2.2.2 Forms Based Authentication Required Response Header	9
2.2.3 HTML Request	9
3 Protocol Details	11
3.1 Common Details	11
3.1.1 Abstract Data Model	11
3.1.2 Timers	11
3.1.3 Initialization	11
3.1.4 Higher-Layer Triggered Events	11
3.1.5 Message Processing Events and Sequencing Rules	11
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	12
3.2.4 Higher-Layer Triggered Events	12
3.2.5 Message Processing Events and Sequencing Rules	12
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 Triggered Events	12
3.3.5 Message Processing Events and Sequencing Rules	12
3.3.6 Timer Events	13
3.3.7 Other Local Events	13
4 Protocol Examples	14
5 Security	16
5.1 Security Considerations for Implementers	16
5.2 Index of Security Parameters	16

6 Appendix A: Product Behavior	17
6.1 SharePoint Login and Return URI	17
7 Change Tracking.....	19
8 Index	20

1 Introduction

This document specifies the Office Forms Based Authentication (OFBA) over the HTTP Protocol (for more information, see [\[RFC2616\]](#)) in addition to other standard authentication mechanisms. This protocol offers protocol clients and servers with HTTP-based-forms authentication when other authentication mechanisms (as specified in [\[RFC4559\]](#) and [\[RFC2617\]](#)) are not available.

1.1 Glossary

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.

[MS-WEBDAVE] Microsoft Corporation, "[Web Distributed Authoring and Versioning Error Extensions Protocol Specification](#)", June 2008.

[MS-WSSHP] Microsoft Corporation, "[HTTP Windows SharePoint Services Headers Protocol Specification](#)", June 2008.

[RFC1945] Berners-Lee, T., Fielding, R., and Frystyk, H., "Hypertext Transfer Protocol -- HTTP/1.0", RFC 1945, May 1996, <http://www.ietf.org/rfc/rfc1945.txt>

[RFC2109] Kristol, D., and Montulli, L., "HTTP State Management Mechanism", RFC 2109, February 1997, <http://www.ietf.org/rfc/rfc2109.txt>

[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>

[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>

[RFC3986] Berners-Lee, T., Fielding, R., and Masinter, L., "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005, <http://www.ietf.org/rfc/rfc3986.txt>

1.2.2 Informative References

[MS-AUTHSO] Microsoft Corporation, "[Windows Authentication Services System Overview](#)", August 2009.

[MS-OCPROTO] Microsoft Corporation, "[Office Client Protocols Overview](#)", June 2008.

[RFC4559] Jaganathan, K., Zhu, L., and Brezak, J., "SPNEGO-based Kerberos and NTLM HTTP Authentication in Microsoft Windows", RFC 4559, June 2006, <http://www.ietf.org/rfc/rfc4559.txt>

1.3 Protocol Overview (Synopsis)

The protocol client connects to a protocol server gated by Office Forms Based Authentication by sending messages via HTTP. The following Sequence Diagram illustrates one possible way of establishing an identity using forms based authentication between a protocol client and a protocol server, entailing three steps.

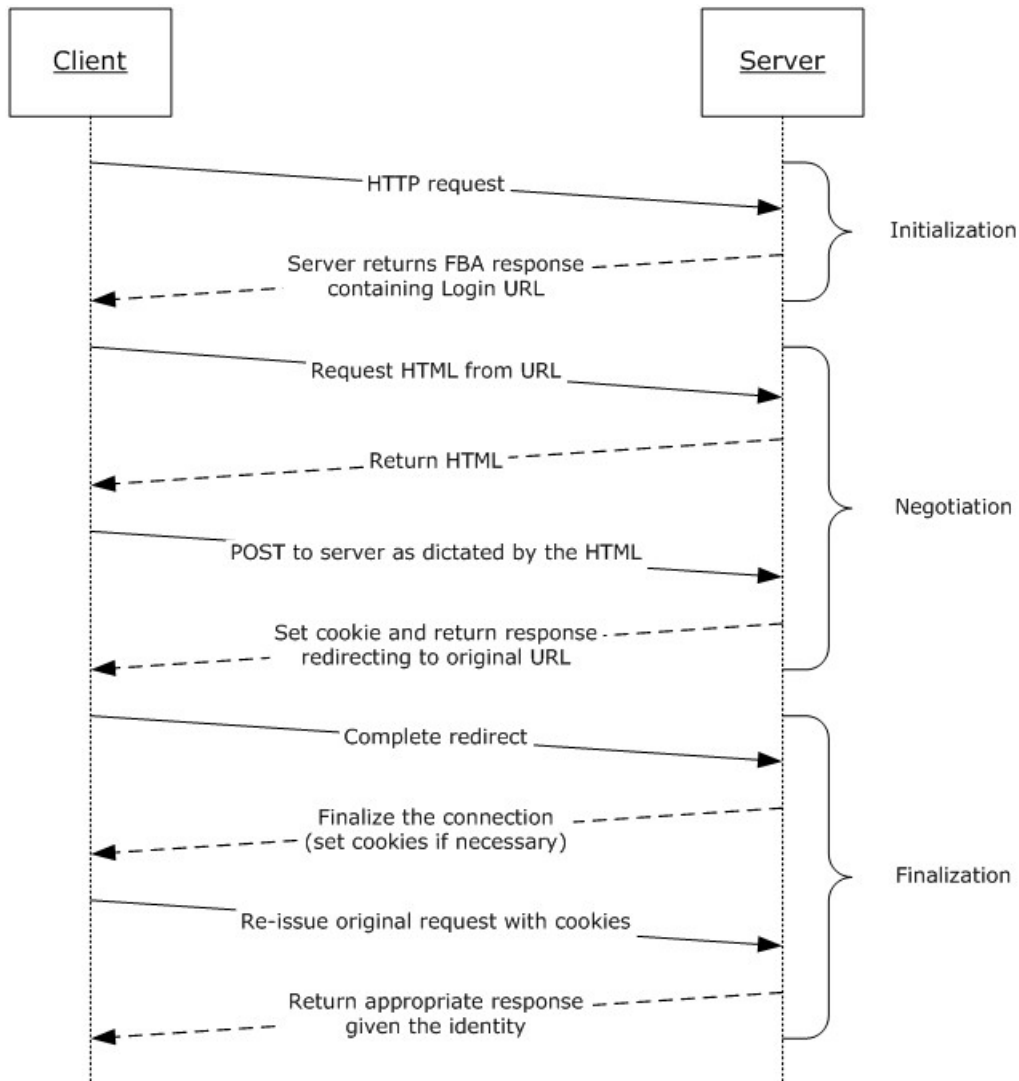


Figure 1: Sequence diagram

The three steps for establishing an identity using forms based authentication between a protocol client and a protocol server are as follows:

Initialization – The protocol client sends an initial request for any transaction between client and the protocol server. The server responds that its authentication method is forms based, as specified in section [2.2.2](#), including the location to which the client should navigate to authenticate. If the server response does not include this location, then the location is assumed to be the location to which the

original request was issued. This response optionally includes the location to which the protocol server will redirect the user upon successfully authenticating the user.

Negotiation – Having determined that the protocol server is capable of establishing an identity using forms based authentication, the client renders the HTML returned from the request to the remote location provided by the server in step (1). Note that the length of this stage is non-deterministic and not specified by this protocol, as the client will continue to follow as many redirects and refreshes as are necessary to successfully establish the identity until the server redirects either to the original URI, or, if specified, the return URI provided by the server in step (1).

Finalization – Once the server redirects the client to the return URI, the protocol client assumes that the identity has been successfully established, and re-issues the original request from step (1). Note that the process of actually establishing the user's identity is not specified by this protocol.

1.4 Relationship to Other Protocols

This protocol depends upon the HTTP transport, as described in [\[RFC2616\]](#). To transfer the authentication state between the client and the server, this protocol is also dependent upon HTTP state management, as described in [\[RFC2109\]](#).

1.5 Prerequisites/Preconditions

Forms based authentication over HTTP assumes the following:

The HTTP Server is configured such that the user's identity is established using forms-based authentication. The user's identity is transferred between the client and the server using HTTP state management as described in [\[RFC2109\]](#).

The client is configured to store and transmit cookies as described in [\[RFC2109\]](#).

1.6 Applicability Statement

Forms based authentication is used in environments where Windows Integrated Authentication methods (basic, digest, SPNEGO-based Kerberos and NTLM HTTP Authentication), which are defined in [\[MS-AUTHSO\]](#) section 2, are not available. Additionally, the protocol client and protocol server must both support Forms Based Authentication.

1.7 Versioning and Capability Negotiation

Versioning and capability is handled by the HTTP protocols described in [\[RFC2617\]](#) (for more information, see [\[RFC2616\]](#)). This protocol has no additional versioning or capability negotiation.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

2.1 Transport

Forms-based authentication over HTTP Protocol messages are carried in the HTTP message headers ([\[RFC2616\]](#) section 4.2) and message body ([\[RFC2616\]](#) section 4.3).

2.2 Message Syntax

The use of forms-based authentication over HTTP Protocol is indicated by an HTTP Response header X-FORMS_BASED_AUTH_REQUIRED. The value of this header is a URI that points to an HTTP-based server. For more details about HTTP headers, see [\[RFC2616\]](#). For more details about URIs, see [\[RFC3986\]](#).

2.2.1 Protocol Discovery Requests

The protocol client will establish an identity with a protocol server based on a specific challenge issued by the client to the server, which identifies the protocol client as a non-browser client application.

To be recognized as a non-browser client that supports this protocol, the protocol client MUST specify either a header ([\[RFC2616\]](#) section 4.2) or a user agent string ([\[RFC1945\]](#) section 10.3) in an HTTP OPTIONS request ([\[RFC2616\]](#) section 9.2). If the protocol client request is not authenticated, the protocol server SHOULD respond based on the criteria specified in order of priority in the following table, but the protocol server MAY ignore the header and use only the user agent string, as specified after the table. <1>

Windows Authentication is described in [\[MS-AUTHSO\]](#) section 2.

Client request	Server response
header contains field-name "X-FORMS_BASED_AUTH_ACCEPTED" and field-value of "f"	If the protocol server supports any type of Windows Authentication, the protocol server MUST NOT respond with a Forms Based Authentication Required response header (section 2.2.2) and MUST respond with a Windows Authentication challenge. If the protocol server does not support any type of Windows Authentication, then the server MUST respond with a Forms Based Authentication Required response header (section 2.2.2)
header does not contain field-name "X-FORMS_BASED_AUTH_ACCEPTED" and user agent string contains "MS Search" followed by "Robot"	If the protocol server supports any type of Windows Authentication, the protocol server MUST NOT respond with a Forms Based Authentication Required response header (section 2.2.2) and MUST respond back with a Windows Authentication challenge. If the protocol server does not support any type of Windows Authentication, then the server MUST respond with a Forms Based Authentication Required response header (section 2.2.2).
header contains field-name "X-FORMS_BASED_AUTH_ACCEPTED" and field-value of "t"	The protocol server MUST respond with a Forms Based Authentication Required response header, as specified in section 2.2.2 .

If the HTTP request sent by the protocol client is not authenticated and does not include the HTTP header "X-FORMS_BASED_AUTH_ACCEPTED" <2>, and the user agent string contains one of the following identifications, the protocol server MUST respond with the Forms Based Authentication Required response header specified in section 2.2.2.

1. "Microsoft Data Access Internet Publishing Provider"
2. "Microsoft-WebDAV-MiniRedir"
3. "non-browser"
4. "MSOffice 12"
5. "Mozilla/4.0 (compatible; MS FrontPage)"

2.2.2 Forms Based Authentication Required Response Header

If the protocol server receives a request for an access-protected object and the request requires a forms-based authentication required response as specified in section 2.2.1, the server MUST respond with a "403 Forbidden" HTTP status code ([RFC2616] section 10.4.4) and an X-FORMS_BASED_AUTH_REQUIRED header as specified in [MS-WSSHP] section 2.2.12. The value of the header MUST be a URI, as specified by [RFC3986], that specifies the protocol server login page. The protocol client will navigate to the login page to establish the user's identity with the protocol server.

The protocol server SHOULD return an HTTP header with **field-name** X-FORMS_BASED_AUTH_RETURN_URL header <3>, as specified in [MS-WSSHP] section 2.2.13. The value of this header contains a URI, as specified by [RFC3986], that specifies the protocol server return page, which the protocol client will use to determine whether authentication succeeded. If the URI is not present, then the protocol client assumes that the URI is the same as the login page specified by the X-FORMS_BASED_AUTH_REQUIRED header. If the URI of the return page is a path, then the path MUST contain a '/' at the end.

The server MAY return an HTTP header with **field-name** X-FORMS_BASED_AUTH_DIALOG_SIZE. The value of this header MUST be formatted as a string that conforms to the following ABNF [RFC4234] grammar:

```
size    = width "x" height
width   = 1*10(DIGIT)
height  = 1*10(DIGIT)
```

width specifies the preferred width, in pixels, of the login dialog.

height specifies the preferred height, in pixels, of the login dialog.

If the dialog size is not specified, then the value "660x495" is used by the protocol client.

Both the login page and the return page MUST point to an HTTP-based server.

2.2.3 HTML Request

Once the protocol client has determined that the user's identity will be established using forms-based authentication, the client MUST issue an HTTP GET ([RFC2616] section 9.3) to the login page. The user agent string ([RFC1945] section 10.3) of this GET request MUST contain:

3 Protocol Details

3.1 Common Details

The Forms Based Authentication Protocol is used to establish a user's identity with a remote protocol server that uses an HTML form to establish that user's identity, for which reason a model that leverages existing HTTP and HTML semantics within the client is useful.

3.1.1 Abstract Data Model

The protocol client relies on the remote protocol server to set the user's identity as one or more HTTP cookies. Once the user's identity is established, the client will then send each cookie with each subsequent HTTP request, per the rules set forth in [\[RFC2109\]](#).

3.1.2 Timers

None.

3.1.3 Initialization

None.

3.1.4 Higher-Layer Triggered Events

None.

3.1.5 Message Processing Events and Sequencing Rules

The Protocol Discovery request **MUST** be sent by the client (for details, see section [2.2.1](#)). The X-FORMS_BASED_AUTH_REQUIRED header and the X-FORMS_BASED_AUTH_RETURN_URL header **MUST** be returned by the server (for details, see section [2.2.2](#)). Client and servers **MUST** be compliant with [\[RFC2616\]](#), [\[RFC2617\]](#), and [\[RFC2109\]](#).

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Client Details

3.2.1 Abstract Data Model

The abstract data model follows that set forth in section [3.1.1](#).

3.2.2 Timers

The client has a single timer used to determine how much time should elapse prior to re-issuing the **Protocol Discovery** request.

The value of this timer is not specified by this protocol, and it is up to the protocol client to choose an optimal value.

3.2.3 Initialization

The protocol is initialized upon receiving the X-FORMS_BASED_AUTH_REQUIRED header from the server.

3.2.4 Higher-Layer Triggered Events

The end-user MAY cause additional requests to be sent to the server, depending upon the contents of the HTML rendered by the client. The number of HTTP requests sent by the protocol client to the protocol server to establish the user's identity are non-deterministic and not specified by this protocol.

3.2.5 Message Processing Events and Sequencing Rules

There are no additional message processing events and sequencing rules beyond those detailed in section [3.1.5](#).

3.2.6 Timer Events

When the timer detailed in section [3.2.2](#) expires, the Protocol Discovery request, detailed in section [2.2.1](#), will first be sent prior to any additional HTTP request made by the client.

3.2.7 Other Local Events

When the remote server issues a redirect to the return page, the client will complete that request, and then allow any other pending HTTP requests to continue. Those HTTP requests MUST contain the cookies, as specified by [\[RFC2109\]](#).

3.3 Server Details

3.3.1 Abstract Data Model

The abstract data model follows that set forth in section [3.1.1](#).

3.3.2 Timers

None.

3.3.3 Initialization

The protocol is initialized upon receiving the Protocol Discovery request from the client.

3.3.4 Higher-Layer Triggered Events

None.

3.3.5 Message Processing Events and Sequencing Rules

There are no additional message processing events and sequencing rules beyond those detailed in section [3.1.5](#).

3.3.6 Timer Events

None.

3.3.7 Other Local Events

None.

4 Protocol Examples

This scenario shows the message exchanges when a protocol client requests an access-protected document from a Forms Based Authentication gated protocol server at the URI: <https://www.contoso.com/dir/document.docx>.

Prior to requesting this document, the client will attempt to determine the capabilities of the server:

```
C: OPTIONS /dir/  
C: User-Agent: MSOffice 12
```

The server issues a response indicating it is forms based authentication capable:

```
S: HTTP/1.1 403 Forbidden  
S: X-FORMS_BASED_AUTH_REQUIRED:  
https://www.contoso.com/fbalogin.aspx?wreply=https://www.contoso.com/OnSuccess.aspx  
S: X-FORMS_BASED_AUTH_RETURN_URL: https://www.contoso.com/OnSuccess.aspx  
S: X-FORMS_BASED_AUTH_DIALOG_SIZE: 800x600.
```

The client will then issue an HTTP request to the header specified in the X-FORMS_BASED_AUTH_REQUIRED URI, requesting HTML that the user can leverage to establish his or her identity:

```
C: GET /fbalogin.aspx?wreply=https://www.contoso.com/OnSuccess.aspx  
C: User-Agent: Mozilla/4.0
```

The server will then reply with an HTML form containing enough logic to establish the user's identity with the server. In this example, the server simply returns a simple form.

```
S: HTTP/1.1 200 OK  
S: <body>  
S: <form name="CredentialForm" method="post"  
S: action="fbalogin.aspx?wreply=https://www.contoso.com/OnSuccess.aspx"  
S: id="Creds">  
S: <table>  
S: <tr><td>Login: </td></tr>  
S: <tr><td>Username: <input name="UsernameTextBox" type="text"  
S: id="UsernameTextBox" </td></tr>  
S: <tr><td>Password: <input name="PasswordTextBox"  
S: type="password"  
S: id="PasswordTextBox" /></td></tr>  
S: <tr><td><input type="submit" name="UsernamePasswordButton"  
S: value="Submit" id="UsernamePasswordButton" /></td></tr>  
S: </table>  
S: </form>  
S: </body>
```

On receipt of this HTML, the client will instantiate a dialog box of the size specified in the initial response to the OPTIONS request (in this example, that size would be 800x600). Once rendered, the rich client will follow the instructions dictated by the HTML form. In this example, assume that

the user entered the credentials "user:pass" for the user name and password, and then clicked the submit button.

```
C: POST /fbaloglein.aspx?wreply=https://www.contoso.com/OnSuccess.aspx
C: User-Agent: Mozilla/4.0
C: UsernameTextBox=user&PasswordTextBox=pass
```

If the user's interactions with the HTML form have allowed the server to establish the user's identity, then the remote server sets the identity as a cookie on the request, and redirects the user back to the return_url specified in the response to the Protocol Discovery Request.

```
S: HTTP/1.1 302 Object Moved
Location: https://www.contoso.com/OnSuccess.aspx
Set-Cookie: Authentication=<server determined hash of the user's identity>
```

Upon seeing the redirect, the client determines that this URI matches that returned in response to the Protocol Discovery request. Because the URIs match, the client assumes success, follows the redirect, and then closes the form that it was using to render the HTML

```
C: GET /OnSuccess.aspx
C: User-Agent: Mozilla/4.0
C: Cookie: Authentication=<server determined hash of the user's identity>
```

The server can then respond with any sort of finalization logic required.

```
S: HTTP/1.1 200 OK
S: Set-Cookie: FooCookie=bar
```

Once this call completes, the client will then run the series of HTTP transactions required to successfully open <https://www.contoso.com/dir/document.docx>. For more information about this series of transactions, see [\[MS-OCPROTO\]](#) section 2.1.2.1.2.

5 Security

5.1 Security Considerations for Implementers

Forms Based Authentication necessarily transmits the user's identity as clear text. Implementers are encouraged to use secure channels, such as HTTPS, to avoid inadvertently exposing the user's identity.

5.2 Index of Security Parameters

None.

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:

- The 2007 Microsoft® Office system
- Microsoft® Office 2010 suites
- Microsoft® SharePoint® Foundation 2010
- Windows® SharePoint® Services 3.0

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 2.2.1:](#) SharePoint 2007 Products and Technologies ignores the header and processes the request based solely on the user agent string, not including "MS Search" followed by "Robot".

[<2> Section 2.2.1:](#) Microsoft Office 2010 and the 2007 Office systemclient applications never send the X-FORMS_BASED_AUTH_ACCEPTED header and always rely on the user agent string to identify themselves as non-browser clients to a protocol server.

[<3> Section 2.2.2:](#) SharePoint 2007 Products and Technologies does not return the X-FORMS_BASED_AUTH_RETURN_URL header. Rather, the protocol server returns the following extended error, as defined in [\[MS-WEBDAVE\]](#) section 2.2.3:

```
X-MSDAVEXT_Error: 917656;  
Access%20denied%2e%20%20Before%20opening%20files%20in%20this%20location%2c%20you%20must%20fir  
st%20browse%20to%20the%20web%20site%20and%20select%20the%20option%20to%20login%20automaticall  
y%2e
```

6.1 SharePoint Login and Return URI

Microsoft® SharePoint 2007 Products and Technologies does not explicitly return to any protocol client the X-FORMS_BASED_AUTH_REQUIRED header to the Protocol Discovery request as detailed in section [2.2.2](#). Rather, the server returns the following extended error, as defined in [\[MS-WEBDAVE\]](#) section 2.2.3:

```
X-MSDAVEXT_Error: 917656;  
Access%20denied%2e%20%20Before%20opening%20files%20in%20this%20location%2c%20you%20must%20fir  
st%20browse%20to%20the%20web%20site%20and%20select%20the%20option%20to%20login%20automaticall  
y%2e
```

Upon receipt of this error, the protocol client SHOULD issue a request to determine the Web URL for the given URL, as defined in [\[MS-WEBDAVE\]](#).

Upon determination of the Web URL, the client SHOULD consider the following URL to be the equivalent of the value for X-FORMS_BASED_AUTH_REQUIRED, as defined in section [2.2.2](#):

```
http://server/weburl/_layouts/Authenticate.aspx?Source=Error.aspx
```

Where

- `Server` is the address of the SharePoint 2007 Products and Technologies server.
- `Weburl` is the value returned from the previous **UrlToWebUrl** request.
- `/_layouts/Authenticate.aspx?Source=Error.aspx` is a hard-coded string.

Additionally, because the server SHOULD return the client to the Error.aspx page on successful authentication, the client SHOULD consider the following URL equivalent to the value of the X-FORMS_BASED_AUTH_RETURN_URL, as defined in section [2.2.2](#):

```
http://server/weburl/_layouts/Error.aspx
```

Where

- `Server` is the address of the SharePoint 2007 Products and Technologies server.
- `Weburl` is the value returned from the previous `UrlToWebUrl` request.
- `/_layouts/Error.aspx` is a hard coded string.

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 the aforementioned Microsoft products' behavior is in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies these Microsoft products do not follow the prescription.

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](#) 11
[overview](#) 11
[server](#) 12
[Applicability](#) 7

C

[Capability negotiation](#) 7
[Change tracking](#) 19
Client
[abstract data model](#) 11
[higher-layer triggered events](#) 12
[initialization](#) 12
[message processing](#) 12
[other local events](#) 12
[overview](#) 11
[sequencing rules](#) 12
[timer events](#) 12
[timers](#) 11

D

Data model - abstract
[client](#) 11
[server](#) 12
Data model – abstract
[overview](#) 11

E

Examples
[overview](#) 14

F

[Fields - vendor-extensible](#) 7
[Forms Based Authentication Required Response Header message](#) 9

G

[Glossary](#) 5

H

Higher-layer triggered events
[client](#) 12
[overview](#) 11
[server](#) 12
[HTML Request message](#) 9

I

[Implementer - security considerations](#) 16
[Index of security parameters](#) 16
[Informative references](#) 5

Initialization
[client](#) 12
[overview](#) 11
[server](#) 12
[Introduction](#) 5

L

Local events
[overview](#) 11
[Login and return URI](#) 17

M

Message processing
[client](#) 12
[overview](#) 11
[server](#) 12
[Message syntax](#) 8
Messages
[Forms Based Authentication Required Response Header](#) 9
[HTML Request](#) 9
[Protocol Discovery Requests](#) 8
[syntax](#) 8
transport ([section 2.1](#) 8, [section 2.2.3](#) 9)

N

[Normative references](#) 5

O

Other local events
[client](#) 12
[server](#) 13
Overview
Protocol Details
[Common](#) 13
[Overview \(synopsis\)](#) 6

P

[Parameters - security index](#) 16
[Preconditions](#) 7
[Prerequisites](#) 7
[Product behavior](#) 17
[Protocol Discovery Requests message](#) 8

Proxy
[overview](#) 11

R

References
[informative](#) 5
[normative](#) 5
[Relationship to other protocols](#) 7

S

Security
[implementer considerations](#) 16
[parameter index](#) 16

Sequencing rules
[client](#) 12
[overview](#) 11
[server](#) 12

Server
[abstract data model](#) 12
[higher-layer triggered events](#) 12
[initialization](#) 12
[message processing](#) 12
[other local events](#) 13
[overview](#) 11
[sequencing rules](#) 12
[timer events](#) 13
[timers](#) 12
[Standards assignments](#) 7

T

Timer events
[client](#) 12
[overview](#) 11
[server](#) 13

Timers
[client](#) 11
[overview](#) 11
[server](#) 12

[Tracking changes](#) 19

Transport ([section 2.1](#) 8, [section 2.2.3](#) 9)

Triggered events - higher-layer
[client](#) 12
[server](#) 12

Triggered events – higher-layer
[overview](#) 11

V

[Vendor-extensible fields](#) 7
[Versioning](#) 7