[MS-XLOGIN]:
Simple Mail Transfer Protocol (SMTP) AUTH LOGIN Extension

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.mspx). 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
04/04/2008	0.1		Initial Availability.
06/27/2008	1.0		Initial Release.
08/06/2008	1.01		Revised and edited technical content.
09/03/2008	1.02		Updated references.
12/03/2008	1.03		Updated IP notice.
04/10/2009	2.0		Updated applicable product releases.
07/15/2009	3.0	Major	Revised and edited for technical content.
11/04/2009	4.0.0	Major	Updated and revised the technical content.
02/10/2010	4.1.0	Minor	Updated the technical content.
05/05/2010	4.1.1	Editorial	Revised and edited the technical content.
08/04/2010	4.1.1	No change	No changes to the meaning, language, or formatting of the technical content.
11/03/2010	4.1.1	No change	No changes to the meaning, language, or formatting of the technical content.
03/18/2011	4.1.1	No change	No changes to the meaning, language, or formatting of the technical content.

Table of Contents

1 Introduction

This document specifies an authentication mechanism that is provided through the **SMTP** Service Extension for Authentication [RFC4954] called AUTH LOGIN.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

Augmented Backus-Naur Form (ABNF)
Secure Sockets Layer (SSL)

The following terms are defined in [MS-OXGLOS]:

base64 encoding
Simple Mail Transfer Protocol (SMTP)
Transport Layer Security (TLS)

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. 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.

[RFC2045] Freed, N., and Borenstein, N., "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996, http://ietf.org/rfc/rfc2045.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

[RFC2821] Klensin, J., "Simple Mail Transfer Protocol", STD 10, RFC 2821, April 2001, http://www.ietf.org/rfc/rfc2821.txt

[RFC4346] Dierks, T., and Rescorla, E., "The Transport Layer Security (TLS) Protocol Version 1.1", RFC 4346, April 2006, http://www.ietf.org/rfc/rfc4346.txt

[RFC4954] Siemborski, R., and Melnikov, A., Eds., "SMTP Service Extension for Authentication", RFC 4954, July 2007, http://www.rfc-editor.org/rfc/rfc4954.txt

[RFC5234] Crocker, D., Ed., and Overell, P., "Augmented BNF for Syntax Specifications: ABNF", STD 68, RFC 5234, January 2008, http://www.ietf.org/rfc/rfc5234.txt

1.2.2 Informative References

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

[MS-OXGLOS] Microsoft Corporation, "Exchange Server Protocols Master Glossary", April 2008.

1.3 Overview

Client applications that connect to the SMTP service can use a number of different authentication mechanisms. The SMTP Service Extension for Authentication [RFC4954] defines a service extension to SMTP as specified in [RFC2821], where a client specifies an authentication method to the server. AUTH LOGIN is one such authentication protocol for SMTP.

1.4 Relationship to Other Protocols

This specification defines an implementation of a protocol using [RFC4954].

1.5 Prerequisites/Preconditions

All prerequisites and preconditions are specified in [RFC2821] and [RFC4954].

1.6 Applicability Statement

This protocol extension does not change the way that [RFC4954] is used.

1.7 Versioning and Capability Negotiation

None.

1.8 Vendor-Extensible Fields

None.

1.9 Standards Assignments

None.

2 Messages

The following sections specify the AUTH LOGIN message syntax and how they are transported.

2.1 Transport

This protocol extension does not change the base transport specified by [RFC2821] or its extension specified by [RFC4954].

2.2 Message Syntax

2.2.1 EHLO Response

Server support for the AUTH LOGIN extension is identified in the AUTH EHLO keyword in the EHLO response, as specified in [RFC4954] section 3. In the following example, "C:" and "S:" indicate lines sent by the client and server respectively.

```
S: 220 SMTP.example.com
C: EHLO client.example.com
S: 250-smtp.example.com  Hello client.example.com
S: 250 AUTH LOGIN
```

The server response of AUTH LOGIN as part of the responses to EHLO indicates support for AUTH LOGIN.

2.2.2 Command and Response ABNF Grammar

This section uses **Augmented Backus-Naur Form (ABNF)** (as specified in [RFC5234]) to define the format of commands and responses used by this document. Note that the values of *username* and *password* are encoded using **base64 encoding** as specified in [RFC2045] before being transmitted.

```
CR                              = %x0D
LF                              = %x0A
SP                              = %x20

username                        = 1*CHAR                        ; Base64-encoded username
password                        = 1*CHAR                        ; Base64-encoded password

auth_login_command              = "AUTH LOGIN" CR LF
auth_login_command_user         = "AUTH LOGIN" SP username CR LF
auth_login_username_challenge   = "334 VXNlcm5hbWU6" CR LF
auth_login_username_response    = username CR LF
auth_login_password_challenge   = "334 UGFzc3dvcmQ6" CR LF
auth_login_password_response    = password CR LF
```

3 Protocol Details

The following sections specify details of the AUTH LOGIN protocol extension, including abstract data models and message processing rules.

3.1 Client Details

3.1.1 Abstract Data Model

This section describes a conceptual model of possible data organization that an implementation maintains to participate in this protocol. The described organization is provided to facilitate the explanation of how the protocol behaves. This 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 abstract data model shown below represents the sequence of messages relative to the client.

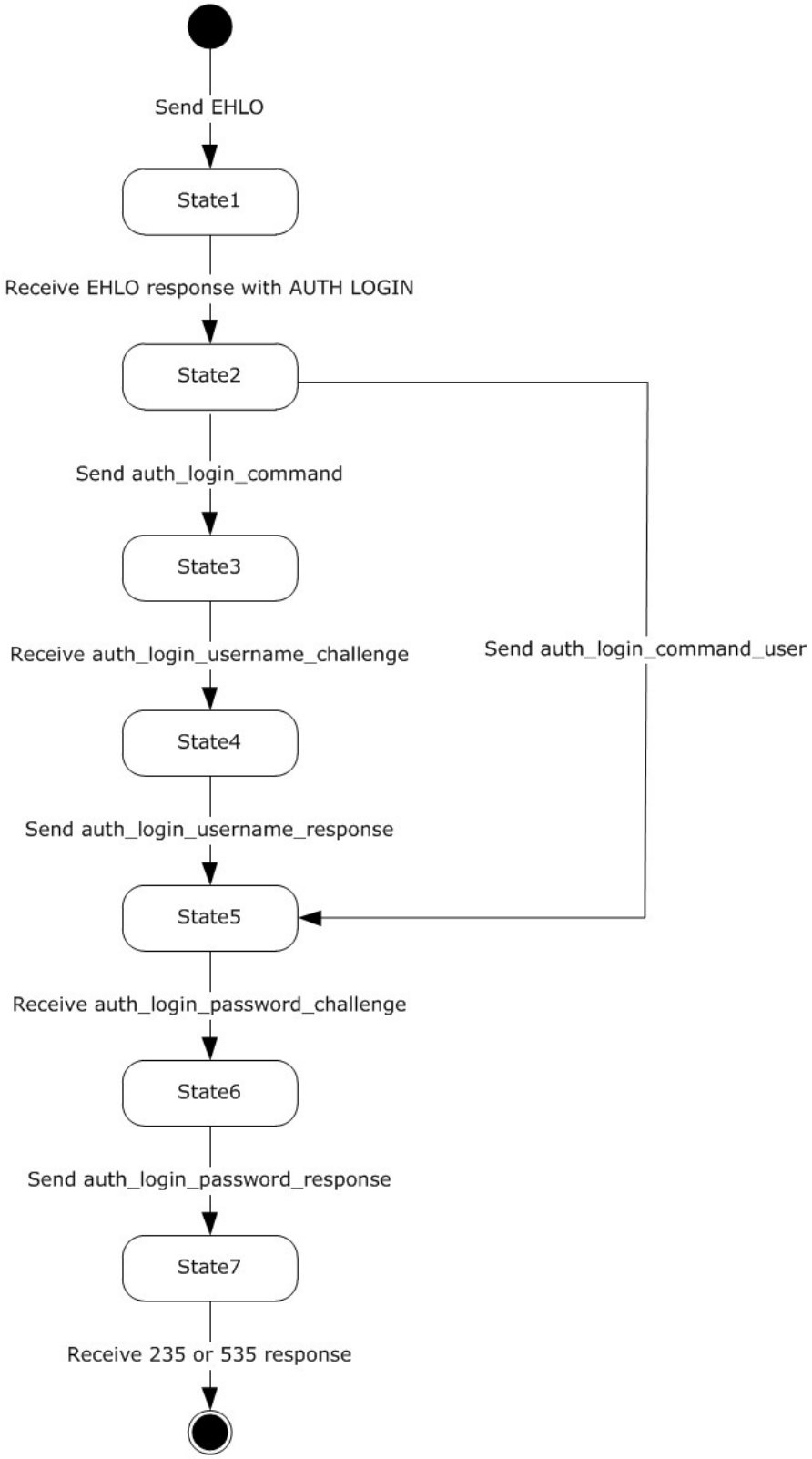

Figure 1: Client abstract data model

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

This protocol does not change message processing events or sequencing rules of messages specified in [RFC4954]. Clients SHOULD ignore any enhanced status codes returned by the server.

3.1.5.1 Sending EHLO

Before requesting authentication, the client SHOULD send an EHLO command and examine the response. If the response does not contain the AUTH LOGIN keyword specified in section 2.2.1, the client SHOULD NOT attempt to use LOGIN authentication.

3.1.5.2 Requesting Simple Login Authentication

To request LOGIN authentication, the client issues the AUTH command with the parameter LOGIN (*auth_login_command* as specified in section 2.2.2).

```
AUTH LOGIN<CR><LF>
```

If AUTH LOGIN is not supported, then the server responds with a 504 error message as specified in [RFC4954] section 4.

If AUTH LOGIN is supported on the server, then the server responds with *auth_login_username_challenge* (as specified in section 2.2.2).

```
334 VXNlcm5hbWU6<CR><LF>
```

The client then responds with *auth_login_username_response* (as specified in section 2.2.2) with the user name to be used for authentication, base64 encoded as specified in [RFC2045]. For example, if the client's user name is "Charlie", then the client responds with the following:

```
Q2hhcmxpZQ==<CR><LF>
```

The server then responds with *auth_login_password_challenge* (as specified in section 2.2.2).

```
334 UGFzc3dvcmQ6<CR><LF>
```

The client then responds with *auth_login_password_response* (as specified in section 2.2.2) with the password to be used for authentication, base64 encoded. For example, if the client's password was "password", then the client would respond with the following:

```
cGFzc3dvcmQ=<CR><LF>
```

If the authentication is complete, then the server issues a 235 reply for success or a 535 reply for failure as specified in [RFC4954].

3.1.5.3 Requesting Login Authentication with Username

To request LOGIN authentication, the client issues the AUTH command with the parameter LOGIN and the user name to be used for authentication, base64 encoded as specified in [RFC2045]. For example, if the client's user name was "Charlie", then the client would initiate AUTH LOGIN as follows (*auth_login_command_user* as specified in section 2.2.2):

```
AUTH LOGIN Q2hhcmxpZQ==<CR><LF>
```

If AUTH LOGIN is not supported, then the server responds with a 504 error message as specified in [RFC4954] section 4.

If AUTH LOGIN is supported on the server, then the server responds with *auth_login_password_challenge* (as specified in section 2.2.2).

```
334 UGFzc3dvcmQ6<CR><LF>
```

The client then responds with *auth_login_password_response* (as specified in section 2.2.2) with the password to be used for authentication, base64 encoded. For example, if the client's password was "password", then the client would respond with the following:

```
cGFzc3dvcmQ=<CR><LF>
```

If the authentication is complete, then the server issues a 235 reply for success or a 535 reply for failure as specified in [RFC4954].

3.1.6 Timer Events

None.

3.1.7 Other Local Events

None.

3.2 Server Details

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 described organization is provided to facilitate the explanation of 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 abstract data model shown below represents the sequence of messages relative to the server.

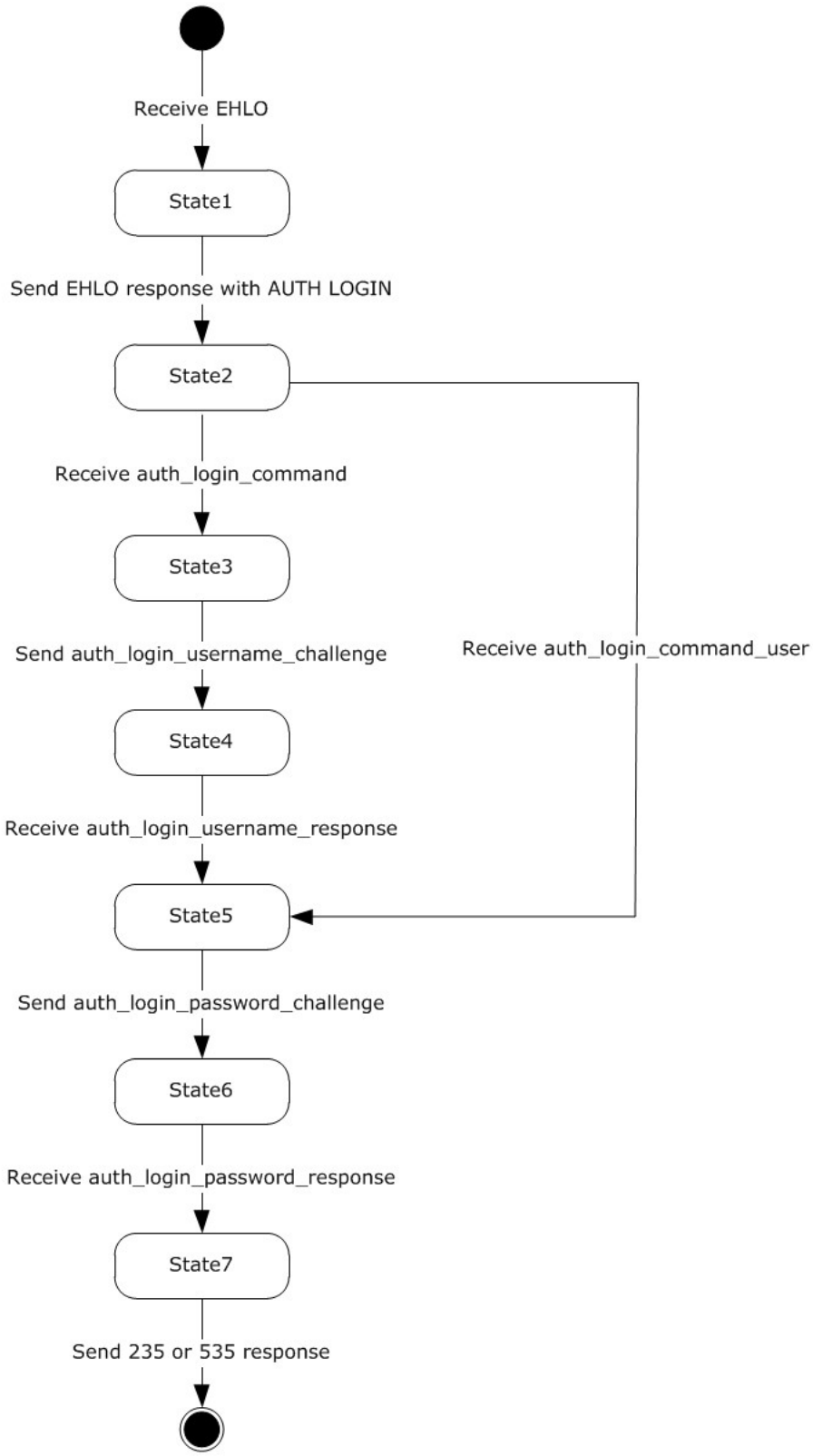

Figure 2: Server abstract data model

3.2.2 Timers

None.

3.2.3 Initialization

None.

3.2.4 Higher-Layer Triggered Events

None.

3.2.5 Message Processing Events and Sequencing Rules

3.2.5.1 Processing EHLO

If the server supports AUTH LOGIN, it MUST include the AUTH keyword with the LOGIN parameter in its response to the EHLO command, as specified in section 2.2.1.

3.2.5.2 Processing AUTH LOGIN

If the server supports AUTH LOGIN, it MUST respond to the *auth_login_command* request with an *auth_login_username_challenge* response, and it MUST respond to the *auth_login_command_user* request with an *auth_login_password_challenge* response.

If the server does not support AUTH LOGIN, it SHOULD respond with a 504 error message as specified in [RFC4954] section 4.

3.2.5.3 Processing Username Response

In order to process an *auth_login_username_response* request, the server MUST have sent an *auth_login_username_challenge* response. If the server receives an *auth_login_username_response* request without having sent an *auth_login_username_challenge* response, the server SHOULD respond with a 500 error message.

If the server receives *auth_login_username_response* request after sending a *auth_login_username_challenge* response, the server SHOULD defer validating the username and SHOULD always send a *auth_login_password_challenge* response.

3.2.5.4 Processing Password Response

In order to process an *auth_login_password_response* request, the server MUST have sent an *auth_login_password_challenge* response. If the server receives an *auth_login_password_response* request without having sent an *auth_login_password_challenge* response, the server SHOULD respond with a 500 error message.

The server SHOULD check that the user indicated by the previous *auth_login_username_response* request is a valid user and that the password sent in the *auth_login_password_response* request is a valid password for that user. If the user and password are valid, then the server SHOULD respond with a 235 response. If the user or password are invalid, then the server SHOULD respond with a 535 response.

3.2.6 Timer Events

None.

3.2.7 Other Local Events

None.

4 Protocol Example

The following is an example of the use of the AUTH LOGIN protocol extension. The example demonstrates SMTP authentication using the AUTH LOGIN protocol.

```
S: 220 SMTP.example.com
C: EHLO client.example.com
S: 250-smtp.example.com  Hello client.example.com
S: 250 AUTH LOGIN
C: AUTH LOGIN
S: 334 VXNlcm5hbWU6
C: Q2hhcmxpZQ==
S: 334 UGFzc3dvcmQ6
C: cGFzc3dvcmQ=
S: 235 authentication successful
```

In this example, a client connects to an SMTP server. The initial response by the server ("220 SMTP.example.com") is the greeting by the server. The client responds with the EHLO command, which is required to activate the extended command set. The server responds with, among other things, an indication of support for AUTH LOGIN. The client then issues the AUTH LOGIN command. The server responds with the first base64-encoded challenge. The client responds with "Q2hhcmxpZQ==", which is the base64-encoded name "Charlie". The server then issues the second base64-encoded challenge, and the client responds with "cGFzc3dvcmQ=", which is the base64-encoded password "password". Successful authentication then results in a server response of "235".

5 Security

5.1 Security Considerations for Implementers

This protocol offers no inherent security mechanisms to protect the user credentials during authentication. Because of this, it is extremely important to only use this protocol when also using a secure communication channel such as **Secure Sockets Layer (SSL)** or **Transport Layer Security (TLS)**. Even when using SSL or TLS, the authentication credentials are available to the SMTP server, where it is possible that they could be recorded for future use. The recommendation, therefore, is to not use this protocol except in circumstances where no other option is available.

In environments where the use of TLS or other external security is mandated, it is strongly recommended that the AUTH LOGIN advertisement be suppressed until a secure channel is negotiated. TLS in particular exhibits this behavior where the SMTP session is restarted after TLS is negotiated. For more information, see [RFC4346].

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:

- Microsoft® Exchange Server 2003
- Microsoft® Exchange Server 2007
- Microsoft® Exchange Server 2010
- Microsoft® Office Outlook® 2003
- Microsoft® Office Outlook® 2007
- Microsoft® Outlook® 2010

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.

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

V