[MS-STANXIMAP]:

Exchange Internet Message Access Protocol (IMAP) Standards Support

This document provides a statement of standards support. It is intended for use in conjunction with the Microsoft technical specifications, publicly available standards specifications, network programming art, and Microsoft distributed systems concepts. It assumes that the reader is either familiar with the aforementioned material or has immediate access to it.

A Standards Support document does not require the use of Microsoft programming tools or programming environments in order to implement the standard. Developers who have access to Microsoft programming tools and environments are free to take advantage of them.

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

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

This document describes the choices made when implementing the Internet Message Access Protocol (IMAP) Standard. It identifies ambiguities and implementer choices and indicates the approach taken in the implementation. The details of the implementation itself are described in the specifications for the relevant protocols or data structures, not in this document.

Revision Summary

Date	Revision History	Revision Class	Comments
7/15/2009	1.0.0	Major	Initial Availability.
10/1/2008	1.1.0	Minor	Updated IP notice.
4/10/2009	2.0.0	Major	Updated applicable product releases.
7/15/2009	3.0.0	Major	Revised and edited technical content.
11/4/2009	3.1.0	Minor	Updated the technical content.
2/10/2010	3.1.0	None	Version 3.1.0 release
8/4/2010	3.2	Minor	Clarified the meaning of the technical content.
11/3/2010	3.3	Minor	Clarified the meaning of the technical content.
3/18/2011	3.4	Minor	Clarified the meaning of the technical content.
8/5/2011	3.4	None	No changes to the meaning, language, or formatting of the technical content.
10/7/2011	4.0	Major	Significantly changed the technical content.
1/20/2012	5.0	Major	Significantly changed the technical content.
4/27/2012	6.0	Major	Significantly changed the technical content.
7/16/2012	6.0	None	No changes to the meaning, language, or formatting of the technical content.
10/8/2012	6.0	None	No changes to the meaning, language, or formatting of the technical content.
2/11/2013	6.0	None	No changes to the meaning, language, or formatting of the technical content.
7/26/2013	6.0	None	No changes to the meaning, language, or formatting of the technical content.
11/18/2013	6.0	None	No changes to the meaning, language, or formatting of the technical content.
2/10/2014	6.0	None	No changes to the meaning, language, or formatting of the technical content.
4/30/2014	6.0	None	No changes to the meaning, language, or formatting of the technical content.
7/31/2014	6.0	None	No changes to the meaning, language, or formatting of the technical content.
10/30/2014	6.0	None	No changes to the meaning, language, or formatting of the technical content.
5/26/2015	7.0	Major	Significantly changed the technical content.
9/14/2015	7.0	None	No changes to the meaning, language, or formatting of the technical content.
6/13/2016	7.0	None	No changes to the meaning, language, or formatting of the

Date	Revision History	Revision Class	Comments
			technical content.
9/14/2016	7.0	None	No changes to the meaning, language, or formatting of the technical content.
2/24/2017	7.0	None	No changes to the meaning, language, or formatting of the technical content.
8/25/2017	8.0	Major	Significantly changed the technical content.
9/19/2017	8.1	Minor	Clarified the meaning of the technical content.

Table of Contents

1 Introduction

This document specifies the level of support provided by Exchange for the Internet Message Access Protocol (IMAP). A client that implements IMAP is able to access and manipulate electronic **mailboxes** on an IMAP server in a way that is functionally equivalent to local folders. The Microsoft Exchange Server IMAP service component processes requests from an IMAP client.

1.1 Glossary

This document uses the following terms:

Augmented Backus-Naur Form (ABNF): A modified version of Backus-Naur Form (BNF), commonly used by Internet specifications. ABNF notation balances compactness and simplicity with reasonable representational power. ABNF differs from standard BNF in its definitions and uses of naming rules, repetition, alternatives, order-independence, and value ranges. For more information, see [RFC5234].

base64 encoding: A binary-to-text encoding scheme whereby an arbitrary sequence of bytes is converted to a sequence of printable ASCII characters, as described in [RFC4648].

Generic Security Services (GSS) API: A programming interface that provides security services to a caller (typically, a communications protocol) in a generic fashion and that allows source-level portability of applications to different environments.

Internet Assigned Numbers Authority (IANA): A central repository for the protocol name and number registries that are used in many Internet protocols.

mailbox: A message store that contains email, calendar items, and other Message objects for a single recipient.

message transfer agent (MTA): An SMTP server that accepts mail from a client or another MTA and delivers the mail or relays it to another MTA.

Multipurpose Internet Mail Extensions (MIME): A set of extensions that redefines and expands support for various types of content in email messages, as described in [RFC2045], [RFC2046], and [RFC2047].

NT LAN Manager (NTLM) Authentication Protocol: A protocol using a challenge-response mechanism for authentication in which clients are able to verify their identities without sending a password to the server. It consists of three messages, commonly referred to as Type 1 (negotiation), Type 2 (challenge) and Type 3 (authentication). For more information, see [MS-NLMP].

SASL: The Simple Authentication and Security Layer, as described in [RFC2222]. This is an authentication mechanism used by the Lightweight Directory Access Protocol (LDAP).

Secure Sockets Layer (SSL): A security protocol that supports confidentiality and integrity of messages in client and server applications that communicate over open networks. SSL uses two keys to encrypt data-a public key known to everyone and a private or secret key known only to the recipient of the message. SSL supports server and, optionally, client authentication using X.509 certificates. For more information, see [X509]. The SSL protocol is precursor to Transport Layer Security (TLS). The TLS version 1.0 specification is based on SSL version 3.0 [SSL3].

Transmission Control Protocol (TCP): A protocol used with the Internet Protocol (IP) to send data in the form of message units between computers over the Internet. TCP handles keeping track of the individual units of data (called packets) that a message is divided into for efficient routing through the Internet.

unique identifier (UID): A pair consisting of a GUID and a version sequence number to identify each resource uniquely. The UID is used to track the object for its entire lifetime through any number of times that the object is modified or renamed.

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

1.2 References

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

1.2.1 Normative References

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

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

[RFC2246] Dierks, T., and Allen, C., "The TLS Protocol Version 1.0", RFC 2246, January 1999, http://www.rfc-editor.org/rfc/rfc2246.txt

[RFC2822] Resnick, P., Ed., "Internet Message Format", RFC 2822, April 2001, http://www.ietf.org/rfc/rfc2822.txt

[RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, March 2003, http://www.rfc-editor.org/rfc/rfc3501.txt

1.2.2 Informative References

None.

1.3 Microsoft Implementations

Microsoft Exchange Server 2007

Microsoft Exchange Server 2010

Microsoft Exchange Server 2013

Microsoft Exchange Server 2016

1.4 Standards Support Requirements

The conformance requirements for [RFC3501] are as follows:

- All required portions of the specification are implemented according to the specification.
- Any recommended portions that are implemented are implemented according to the specification.
- Any optional portions that are implemented are implemented according to the specification.

The following table lists the sections of [RFC3501] that are considered normative and the sections that are considered informative.

Section(s)	Normative/Informative
1	Informative
2 - 7	Normative
8	Informative
9	Normative
10 - 12	Informative

1.5 Notation

The following notations are used to identify clarifications in section 2.2.

Notation	Explanation
C####	This notation identifies a clarification of ambiguity in the target specification. This includes imprecise statements, omitted information, discrepancies, and errata. This does not include data formatting clarifications.
V####	This notation identifies an intended point of variability in the target specification such as the use of MAY, SHOULD, or RECOMMENDED. This does not include extensibility points.
E####	Because the use of extensibility points, such as optional implementation-specific data, could impair interoperability, this notation identifies such points in the target specification.

2 Standards Support Statements

2.1 Normative Variations

The following subsections detail the normative variations from [RFC3501].

2.1.1 [RFC3501] Section 2.1, Port 143

The specification states: "When **TCP** is used, an IMAP4rev1 server listens on port 143."

By default, Microsoft Exchange Server uses port 143 for TCP connections and port 993 for SSL connections. However, Microsoft Exchange can be configured to use any port.

2.1.2 [RFC3501] Section 2.3.1.1, Unique Identifiers MUST NOT Change During Session

The specification states that the unique identifier of a message MUST NOT change during the session.

Microsoft Exchange assigns a new **UID** to a revised message. (A message can be changed by another protocol and, under certain conditions, the revised message replaces the existing message.)

2.1.3 [RFC3501] Section 2.3.1.1, Combination of Mailbox Name, UIDVALIDITY, and UID MUST Refer to Single, Immutable message

The specification states: "The combination of mailbox name, UIDVALIDITY, and UID must refer to a single immutable message on that server forever. In particular, the internal date, [RFC2822] size, envelope, body structure, and message texts (RFC822, RFC822.HEADER, RFC822.TEXT, and all BODY[...] fetch data items) must never change."

Although Microsoft Exchange adheres to this rule, other protocols have access to these messages, and some of these protocols modify message properties such as the message body. Changes to the message in this way result in a new **UID** value for the message.

2.1.4 [RFC3501] Section 7.2.1, Server MUST Support STARTTLS, LOGINDISABLED, and AUTH=PLAIN Capabilities

The specification states that the server implementation MUST support the STARTTLS, the LOGINDISABLED, and the AUTH=PLAIN capabilities.

When the client uses **SSL** to connect to the server, Microsoft Exchange does not support the LOGINDISABLED capability by default, but it can be configured to do so.

2.1.5 [RFC3501] Section 9, ABNF Rules in General

The specification states that **ABNF** rules MUST be strictly followed.

Microsoft Exchange Server 2007, Microsoft Exchange Server 2010, and Microsoft Exchange Server 2010 Service Pack 1 (SP1) do not strictly follow the rules when sending responses or when parsing commands from the client.

Microsoft Exchange Server 2010 Service Pack 2 (SP2), Microsoft Exchange Server 2013, and Microsoft Exchange Server 2016 strictly follow the rules when sending responses but do not strictly follow the rules when parsing commands from the client.

2.1.6 [RFC3501] Section 9, Rule Regarding Spaces

The specification states: "In all cases, SP refers to exactly one space. It is NOT permitted to substitute TAB, insert additional spaces, or otherwise treat SP as being equivalent to LWSP."

Microsoft Exchange strictly follows the rules when sending responses. When parsing a command from the client, Microsoft Exchange accepts a TAB character as a delimiter of the command itself, but allows only one space in all other places.

2.1.7 [RFC3501] Section 11.1, Server MUST Implement the TLS_RSA_WITH_RC4_128_MD5 Cipher Suite

The specification states that the server MUST implement the TLS_RSA_WITH_RC4_128_MD5 cipher suite.

Microsoft Exchange does not implement the TLS_RSA_WITH_RC4_128_MD5 cipher suite and, instead, relies on the operating system to provide the implementation.

2.2 Clarifications

The following subsections identify clarifications relative to [RFC3501].

Unless otherwise stated, the specified products conform to all SHOULD and RECOMMENDED behavior as specified in [RFC3501]. The term "can" is used throughout [RFC3501] and is interpreted to indicate optional behavior.

2.2.1 [RFC3501] Section 2.2.1, Client Protocol Sender and Server Protocol Receiver

C0001:

The specification states that each client command is prefixed with an identifier, called a tag, but does not make a specific requirement on format. Later in the specification (section 9), the syntax is explicitly stated.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange does not enforce any particular format.

V0001:

The specification states that a different tag is generated by the client for each command.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange accepts repeated use of the same tag in subsequent commands.

2.2.2 [RFC3501] Section 2.2.2, Server Protocol Sender and Client Protocol Receiver

V0002:

The specification states: "Server data MAY be sent as a result of a client command, or MAY be sent unilaterally by the server."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange sends data to the client unilaterally.

V0003:

The specification states: "Servers SHOULD enforce the syntax outlined in this specification strictly. Any client command with a protocol syntax error, including (but not limited to) missing or extraneous spaces or arguments, SHOULD be rejected, and the client given a BAD server completion response."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange is liberal in parsing the spaces in commands. For more details, see section 2.1.6 in this document.

2.2.3 [RFC3501] Section 2.3.1.1, Unique Identifier (UID) Message Attribute

V0004:

The specification states that the unique identifier of a message SHOULD NOT change between sessions.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange assigns a new **UID** to a revised message. (A message can be changed by another protocol and, under certain conditions, the revised message replaces the existing message.)

2.2.4 [RFC3501] Section 2.3.2, Flags Message Attribute

E0001:

The specification states that the server can define keywords. (A keyword is a non-system flag.)

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange defines the $MDNSent keyword, which is set on the message when the client sends a message delivery notification (MDN).

V0005:

The specification states: "Servers MAY permit the client to define new keywords in the mailbox."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange does not support client-defined keywords.

2.2.5 [RFC3501] Section 2.3.4, [RFC2822] Size Message Attribute

V0006:

The specification defines the [RFC2822] size message attribute as the number of octets in the message.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

By default, Microsoft Exchange calculates the [RFC2822] size based on the amount of storage space that the message actually occupies. However, Microsoft Exchange can be configured to base the calculation of the [RFC2822] size on the exact **MIME** size of the message.

2.2.6 [RFC3501] Section 4.3.1, 8-bit and Binary Strings

V0007:

The specification states that implementations MAY transmit 8-bit or multi-octet characters in literals, but SHOULD do so only when the **Internet Assigned Numbers Authority (IANA)** registered character set is identified.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange does not transmit 8-bit or multi-octet characters.

2.2.7 [RFC3501] Section 5.1, Mailbox Naming

V0008:

The specification takes no position on case-sensitivity in non-INBOX mailbox names.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange is case-insensitive regarding non-INBOX mailbox names.

V0009:

The specification states: "Any character which is one of the atom-specials will require that the mailbox name be represented as a quoted string or literal."

Exchange 2007

Microsoft Exchange returns a literal for a mailbox name that includes the backslash character ("\"), but does not return a literal for a mailbox name that includes other atom-specials.

Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange returns a literal for a mailbox name that includes the backslash character ("\") or the double-quote character ("""), and returns a quoted string for a mailbox name that includes other atom-specials.

V0010:

The specification states: "Although the list-wildcard characters ('%' and '*') are valid in a mailbox name, it is difficult to use such mailbox names with the LIST and LSUB commands due to the conflict with wildcard interpretation."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange allows a mailbox name to contain a wildcard character and other special characters (such as atom-specials), provided that the characters are escaped.

V0011:

The specification states: "Usually, a character (determined by the server implementation) is reserved to delimit levels of hierarchy."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange uses the forward slash ("/") as the hierarchy delimiter.

V0012:

The specification states: "Two characters, '#' and '&', have meanings by convention, and should be avoided except when used in that convention."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange allows a mailbox name to contain "&", provided that the "&" is encoded as specified in section 5.1.3 of the specification. An unescaped "&" would be treated as a shift to the modified **base64 encoding**, as described in section 5.1.3 of the specification. Microsoft Exchange allows unescaped "#" in folder names.

C0002:

The specification does not impose any limitations on mailbox hierarchy depth.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Hierarchy depth is limited to 31 levels.

C0003:

The specification does not impose any limitations on the length of a mailbox name.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Mailbox names are limited to 250 characters. If a mailbox exists on the server with a name that contains more than 250 characters, the name will be truncated when the mailbox is retrieved.

2.2.8 [RFC3501] Section 5.2, Mailbox Size and Message Status Updates

V0013:

The specification states that agents other than the server MAY add messages to the mailbox, change the flags of the messages in the mailbox, or even remove messages from the mailbox.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange allows non-IMAP protocols to add messages, change flags of messages, and remove messages.

2.2.9 [RFC3501] Section 5.3, Response When No Command in Progress

V0014:

The specification states: "Server implementations are permitted to send an untagged response (except for EXPUNGE) while there is no command in progress. Server implementations that send such responses MUST deal with flow control considerations. Specifically, they MUST either (1) verify that the size of the data does not exceed the underlying transport's available window size, or (2) use non-blocking writes."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange can send an untagged response when there is no command in progress. Microsoft Exchange has mechanisms to manage flow control. Any untagged responses that Microsoft Exchange sends are brief and, therefore, fit into most **message transfer agent (MTA)** windows.

2.2.10 [RFC3501] Section 5.4, Autologout Timer

V0015:

The specification states: "If a server has an inactivity autologout timer, the duration of that timer MUST be at least 30 minutes."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange has an inactivity autologout timer with a default duration of 30 minutes, but can be configured to use a duration of less than 30 minutes.

E0002:

The specification does not describe any other required or optional autologout timers.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange implements an unauthenticated timer, which limits the duration of an unauthenticated session. The default duration of the unauthenticated timer is 60 seconds, but Microsoft Exchange can be configured to use a duration of less than 60 seconds. The receipt of any command from the client during that interval does not reset the unauthenticated timer.

2.2.11 [RFC3501] Section 5.5, Multiple Commands in Progress

V0016:

The specification states that a server MAY begin processing another command before processing the current command to completion, subject to ambiguity rules.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange does not begin processing another command before processing the current command to completion. (Microsoft Exchange processes commands serially.)

V0017:

The specification states: "If the server detects a possible ambiguity, it MUST execute commands to completion in the order given by the client."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange processes commands serially and, therefore, does not need to deal with ambiguities.

2.2.12 [RFC3501] Section 6.2, Client Commands — Not Authenticated State

V0018:

The specification states that server implementations MAY allow access to certain mailboxes without establishing authentication.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange does not allow access to mailboxes without authentication.

V0019:

The specification states that a client can access mailboxes without establishing authentication by using either the ANONYMOUS authenticator or the LOGIN command with a user name of "anonymous".

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange does not support either the ANONYMOUS authenticator or the LOGIN command with an anonymous user name.

2.2.13 [RFC3501] Section 6.2.1, STARTTLS Command

V0020:

The specification states: "The server MAY advertise different capabilities after STARTTLS."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange sends capabilities only in response to a CAPABILITY command from the client. Microsoft Exchange does not send capabilities automatically by using the CAPABILITY response code. For more details, see sections 2.2.14, 2.2.15, and 2.2.31 of this document.

2.2.14 [RFC3501] Section 6.2.2, AUTHENTICATE Command

V0021:

The specification states: "If the server supports the requested authentication mechanism, it performs an authentication protocol exchange to authenticate and identify the client. It MAY also negotiate an OPTIONAL security layer for subsequent protocol interactions."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange does not support negotiation of an OPTIONAL security layer.

E0003:

The specification states that the server is not required to implement any authentication mechanisms other than the PLAIN authentication mechanism.

Exchange 2007

In addition to the PLAIN authentication mechanism, Microsoft Exchange implements the following authentication mechanisms:

- **NTLM**
- **GSSAPI** (also called Kerberos)

Exchange 2010, Exchange 2013, Exchange 2016

In addition to the PLAIN authentication mechanism, Microsoft Exchange implements the GSSAPI authentication mechanism.

V0022:

The specification states that server sites SHOULD NOT use any configuration that permits a plaintext password mechanism without a protection mechanism against password snooping.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

By default, Microsoft Exchange does not permit plaintext password authentication when connecting insecurely, but can be configured to allow plaintext password authentication without protection against password snooping.

E0004:

The specification states that servers SHOULD implement additional **SASL** mechanisms that do not use plaintext passwords.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

For details, see E0005 in this section.

V0023:

The specification states: "A server MAY include a CAPABILITY response code in the tagged OK response of a successful AUTHENTICATE command in order to send capabilities automatically."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange does not include a CAPABILITY response code in its response to a successful AUTHENTICATE command. Microsoft Exchange sends capabilities only in response to a CAPABILITY command from the client. For more details, see sections 2.2.13, 2.2.15, and 2.2.31 of this document.

2.2.15 [RFC3501] Section 6.2.3, LOGIN Command

V0024:

The specification states: "A server MAY include a CAPABILITY response code in the tagged OK response to a successful LOGIN command in order to send capabilities automatically."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange does not include a CAPABILITY response code in its response to a successful LOGIN command. Microsoft Exchange sends capabilities only in response to a CAPABILITY command from the client. For more details, see sections 2.2.13, 2.2.14, and 2.2.31 of this document.

V0025:

The specification states: “Server sites SHOULD NOT use any configuration which permits the LOGIN command without such a protection mechanism against password snooping.”

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

By default, Microsoft Exchange does not permit plaintext password authentication when connecting insecurely, but can be configured to allow plaintext password authentication without protection against password snooping.

2.2.16 [RFC3501] Section 6.3.3, CREATE Command

V0026:

The specification states: "If the server's hierarchy separator character appears elsewhere in the name, the server SHOULD create any superior hierarchical names that are needed for the CREATE command to be successfully completed."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange creates all superior hierarchical names, wherever the server's hierarchy separator character appears in the name, to allow successful completion of the CREATE command.

2.2.17 [RFC3501] Section 6.3.4, DELETE Command

V0027:

The specification states: "It is permitted to delete a name that has inferior hierarchical names and does not have the \Noselect mailbox name attribute."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange does not allow a mailbox to be deleted if it has inferior hierarchical names.

2.2.18 [RFC3501] Section 6.3.5, RENAME Command

V0028:

The specification states: "If the server's hierarchy separator character appears in the name, the server SHOULD create any superior hierarchical names that are needed for the RENAME command to complete successfully."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange does not support the server’s hierarchy separator character in the new name and it returns a BAD response in that case.

2.2.19 [RFC3501] Section 6.3.6, SUBSCRIBE Command

V0029:

The specification states: "A server MAY validate the mailbox argument to SUBSCRIBE to verify that it exists."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange does not validate that the mailbox exists.

2.2.20 [RFC3501] Section 6.3.8, LIST Command

V0030:

The specification prescribes rules that apply to mailbox names that are returned in the LIST response.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

If a mailbox name contains the backslash character ("\"), then Microsoft Exchange returns the mailbox name as a literal preceded by a <length> field.

V0031:

The specification states: "If the reference argument is not a level of mailbox hierarchy (that is, it is a \NoInferiors name), and/or the reference argument does not end with the hierarchy delimiter, it is implementation-dependent how this is interpreted."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange simply returns "LIST completed" in this case.

V0032:

The specification states that server implementations are permitted to hide otherwise accessible mailboxes from the wildcard characters.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange does not hide any mailboxes from wildcard characters.

2.2.21 [RFC3501] Section 6.3.9, LSUB Command

V0033:

The specification prescribes rules that apply to mailbox names that are returned in the LSUB response.

Exchange 2007

If a mailbox name contains the backslash character ("\"), then Microsoft Exchange returns the mailbox name as a literal preceded by a <length> field.

Exchange 2010, Exchange 2013, Exchange 2016

If a mailbox name contains the backslash character ("\") or the double-quote character ("""), then Microsoft Exchange returns the mailbox name as a literal preceded by a <length> field.

V0034:

The specification states: "The returned untagged LSUB response MAY contain different mailbox flags from a LIST untagged response."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange supports an untagged LSUB response containing mailbox flags that are different from those in an untagged LIST response. The flags differ according to the mailbox's existing flags.

2.2.22 [RFC3501] Section 6.3.11, APPEND Command

V0035:

The specification states: "If the mailbox is currently selected, the normal new message actions SHOULD occur. Specifically, the server SHOULD notify the client immediately via an untagged EXISTS response."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

If the mailbox is currently selected, then Microsoft Exchange does not immediately send an untagged EXISTS response. Instead, Microsoft Exchange sends an untagged EXISTS response upon successful completion of APPEND within the currently selected mailbox.

2.2.23 [RFC3501] Section 6.4.1, CHECK Command

V0036:

The specification defines the CHECK command.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange implements the CHECK command as identical to the NOOP command, as specified in [RFC3501] section 6.1.2.

2.2.24 [RFC3501] Section 6.4.4, SEARCH Command

V0037:

The specification states: "US-ASCII MUST be supported; other IANA character sets MAY be supported."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange supports only the US-ASCII character set.

V0038:

The specification states that the tagged NO response SHOULD contain the BADCHARSET response code, which MAY list the IANA character sets that are supported by the server.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange includes the BADCHARSET response code followed by (US-ASCII).

V0039:

The specification defines the KEYWORD <flag> search key as: "Messages with the specified keyword flag set."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange supports only the $MDNSent keyword flag for the KEYWORD search key. For more details, see section 2.2.4 of this document.

V0040:

The specification defines the LARGER <n> search key as: "Messages with an [RFC2822] size larger than the specified number of octets."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

If the [RFC2822] size of the message is known, then Microsoft Exchange compares the [RFC2822] size to the specified number of octets; otherwise, Microsoft Exchange compares the stored-message size (this is the amount of storage space that the message actually occupies) to the specified number of octets. Microsoft Exchange can be configured to evaluate the [RFC2822] size in all circumstances.

V0041:

The specification defines the SMALLER <n> search key as: "Messages with an [RFC2822] size smaller than the specified number of octets."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

If the [RFC2822] size of the message is known, then Microsoft Exchange compares the [RFC2822] size with the specified number of octets; otherwise, Microsoft Exchange compares the stored-message size (this is the amount of storage space that the message actually occupies) of the message to the specified number of octets. Microsoft Exchange can be configured to evaluate the [RFC2822] size in all circumstances.

V0042:

The specification defines the UNKEYWORD <flag> search key as: "Messages that do not have the specified keyword flag set."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange supports only the $MDNSent keyword flag for the UNKEYWORD search key. For more details, see section 2.2.4 of this document.

V0043:

The specification states: "Searching criteria consist of one or more search keys."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange supports recursive grouping of criteria as follows:

- There can be as many as ten levels of recursion.
- Within each recursion, any number of search keys can be used to specify the criteria.

- Parentheses are used to delimit the levels of recursion. For example, four levels of recursion will have the following form: (criteria 1 (OR criteria 2 (OR criteria 3 (OR criteria 4))))

2.2.25 [RFC3501] Section 6.4.5, FETCH Command

V0044:

The specification states that some data items, identified in the formal syntax under the msg-att-dynamic rule, MAY change, either as a result of a STORE command, or due to external events.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange allows such data items to change either as a result of a STORE command or due to external events.

V0045:

The specification describes the RFC822.SIZE data item as: "The [RFC2822] size of the message."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

If the [RFC2822] size of the message is known, then Microsoft Exchange returns that value; otherwise, Microsoft Exchange returns the stored-message size (this is the amount of storage space that the message actually occupies). Microsoft Exchange can be configured to return the [RFC2822] size in all circumstances.

2.2.26 [RFC3501] Section 6.4.6, STORE Command

V0046:

The specification states that, regardless of whether the .SILENT suffix was used in the data item name, the server SHOULD send an untagged FETCH response if a message's flags are changed by an external source.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange never sends an untagged FETCH response if a message's flags are changed by an external source.

2.2.27 [RFC3501] Section 6.5, Client Commands — Experimental/Expansion

V0047:

The specification describes how to define an experimental command or any command that is not part of the specification.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange does not define any such commands.

2.2.28 [RFC3501] Section 7.1, Server Responses — Status Responses

V0048:

The specification states that status responses (OK, NO, BAD, PREAUTH, and BYE) MAY include an OPTIONAL response code.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange includes optional response codes for certain status responses.

V0049:

The specification defines the ALERT response code and the PARSE response code.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange does not implement the ALERT response code and the PARSE response code.

2.2.29 [RFC3501] Section 7.1.1, OK Response

V0050:

The specification states: "The untagged form indicates an information-only message; the nature of the information MAY be indicated by a response code."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange sends an untagged OK response that includes human-readable text with no response code.

2.2.30 [RFC3501] Section 7.1.4, PREAUTH Response

V0051:

The specification defines the PREAUTH response.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange does not implement the PREAUTH response.

2.2.31 [RFC3501] Section 7.2.1, CAPABILITY Response

V0052:

The specification states: "A server MAY send capabilities automatically, by using the CAPABILITY response code in the initial PREAUTH or OK responses, and by sending an updated CAPABILITY response code in the tagged OK response as part of a successful authentication."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange does not use the CAPABILITY response code to automatically send capabilities to the client. Microsoft Exchange sends capabilities to the client only when the client requests capabilities by using the CAPABILITY command. For more details, see sections 2.2.13, 2.2.14, and 2.2.15 of this document.

2.2.32 [RFC3501] Section 7.2.2, LIST Response

V0053:

The specification defines four name attributes for the LIST response.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange uses the \Marked and the \Noselect name attributes. Microsoft Exchange does not use the \Noinferiors and the \Unmarked name attributes.

2.2.33 [RFC3501] Section 7.2.6, FLAGS Response

V0054:

The specification states that flags other than the system flags can also exist, depending on server implementation.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

For more details, see section 2.2.4 of this document.

2.2.34 [RFC3501] Section 7.4.1, EXPUNGE Response

V0055:

The specification states that the server does not need to send an EXISTS response after an untagged EXPUNGE response decrements the number of messages in the mailbox.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Upon successful completion of the EXPUNGE command, Microsoft Exchange sends an EXISTS response with the updated size of the mailbox.

2.2.35 [RFC3501] Section 7.4.2, FETCH Response

V0056:

The specification states: "Extension data is never returned with the BODY fetch, but can be returned with a BODYSTRUCTURE fetch."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange returns extension data as part of a BODYSTRUCTURE fetch.

V0057:

The specification states that a server can return a NIL envelope member in the case where the Date, Subject, In-Reply-To, or Message-ID header line are present but empty.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

If the Date, Subject, In-Reply-To, or Message-ID header line are present but empty, Microsoft Exchange returns a NIL envelope member.

2.2.36 [RFC3501] Section 7.5, Server Responses – Command Continuation Request

V0058:

The specification states that the command continuation request response "is also used if an argument to any command is a literal."

Exchange 2007, Exchange 2010

Microsoft Exchange does not return the command continuation request response when a client issues a UID SEARCH command with a literal.

Exchange 2010 SP2, Exchange 2013, Exchange 2016

Microsoft Exchange returns the command continuation request response when a client sends a literal argument.

2.2.37 [RFC3501] Section 11.1, STARTTLS Security Considerations

V0059:

The specification states that the server SHOULD implement the TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA cipher suite, as specified in [RFC2246].

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange does not implement the TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA cipher suite and, instead, relies on the operating system to provide the implementation.

V0060:

The specification states that all cipher suites other than TLS_RSA_WITH_RC4_128_MD5 and TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA are optional.

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange does not implement any optional cipher suites and, instead, relies on the operating system to provide the implementation of any optional cipher suites.

2.2.38 [RFC3501] Section 11.2, Other Security Considerations

V0061:

The specification states: "A server SHOULD have mechanisms in place to limit or delay failed AUTHENTICATE/LOGIN attempts."

Exchange 2007, Exchange 2010, Exchange 2013, Exchange 2016

Microsoft Exchange allows four failed attempts before it drops the session. However, Microsoft Exchange does not have any cross-session limits.

2.3 Error Handling

Unless otherwise specified in the previous sections, Microsoft Exchange handles errors according to the following:

- Invalid property values and invalid parameter values are ignored.
- Invalid components are ignored.

2.4 Security

None.

3 Change Tracking

This section identifies changes that were made to this document since the last release. Changes are classified as Major, Minor, or None.

The revision class **Major** means that the technical content in the document was significantly revised. Major changes affect protocol interoperability or implementation. Examples of major changes are:

- A document revision that incorporates changes to interoperability requirements.
- A document revision that captures changes to protocol functionality.

The revision class **Minor** means that the meaning of the technical content was clarified. Minor changes do not affect protocol interoperability or implementation. Examples of minor changes are updates to clarify ambiguity at the sentence, paragraph, or table level.

The revision class **None** means that no new technical changes were introduced. Minor editorial and formatting changes may have been made, but the relevant technical content is identical to the last released version.

The changes made to this document are listed in the following table. For more information, please contact dochelp@microsoft.com.

Section	Description	Revision class
2.2.16 [RFC3501] Section 6.3.3, CREATE Command	Clarified the condition for creating all superior hierarchical names.	Minor
2.2.16 [RFC3501] Section 6.3.3, CREATE Command	Clarified description of product behavior when hierarchy separator character appears in the name.	Minor
2.2.23 [RFC3501] Section 6.4.1, CHECK Command	Updated reference section for NOOP command.	Minor

4 Index