

mailmaint  
Internet-Draft  
Updates: RFC8474 (if approved)  
Intended status: Standards Track  
Expires: 6 May 2026

M. De Gennaro  
Stalwart Labs  
2 November 2025

IMAP OBJECTID Account Identifier extension  
draft-degennaro-imap-objectid-accountid-00

## Abstract

This document extends the IMAP OBJECTID extension (RFC 8474) to support account identifiers, enabling IMAP servers to provide account-level context for mailboxes when multiple accounts are accessible through a single IMAP connection. This extension is particularly relevant for environments where IMAP mailboxes include shared mailboxes from multiple JMAP accounts, as defined in the JSON Meta Application Protocol (JMAP). By introducing the ACCOUNTID object identifier, this specification ensures that mailboxes can be uniquely identified across different accounts in both IMAP and JMAP contexts, thereby facilitating seamless interoperability between the two protocols in multi-account scenarios.

## Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 6 May 2026.

## Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Notational Conventions . . . . .	4
2. ACCOUNTID Object Identifier . . . . .	4
2.1. New Response Code for CREATE . . . . .	5
2.2. New OK Untagged Response for SELECT and EXAMINE . . . . .	6
2.3. New Attribute for STATUS . . . . .	7
3. Implementation Considerations . . . . .	8
4. Security Considerations . . . . .	9
4.1. Account Identifier Exposure . . . . .	9
4.2. Cross-Account Information Leakage . . . . .	9
4.3. Consistency with JMAP Authentication . . . . .	9
4.4. Digest-Based Identifier Generation . . . . .	10
4.5. Privacy in Multi-Tenant Environments . . . . .	10
4.6. Other Considerations . . . . .	10
5. IANA Considerations . . . . .	10
5.1. IMAP Capabilities Registry . . . . .	10
5.2. IMAP Response Codes Registry . . . . .	10
6. References . . . . .	10
6.1. Normative References . . . . .	10
6.2. Informative References . . . . .	11
Appendix A. Changes . . . . .	12
Author's Address . . . . .	12

## 1. Introduction

The IMAP OBJECTID extension, defined in [RFC8474], provides persistent identifiers for mailboxes (MAILBOXID) and messages (EMAILID and THREADID) within an IMAP server. These identifiers enable clients to efficiently track and reference resources even when they are moved or renamed, thereby reducing the need for redundant data synchronization operations.

The JMAPACCESS extension for IMAP, specified in [RFC9698], establishes a mechanism by which IMAP clients can discover that the messages and mailboxes accessible via IMAP are also available through JMAP (the JSON Meta Application Protocol, as defined in [RFC8620]). When an IMAP server advertises the JMAPACCESS capability, it asserts

that mailboxes and messages share the same object identifiers across both protocols. This allows clients to migrate gradually to JMAP or to leverage JMAP-specific features while continuing to use IMAP as their primary access method.

The JMAP Email Delivery Push extension [I-D.ietf-jmap-emailpush] extends JMAP's push notification capabilities to deliver EmailPush objects containing specific email properties directly to clients when new messages arrive. These push notifications include the accountId property, which identifies the JMAP account to which the delivered email belongs. This account-level identification is essential for clients managing multiple accounts, as it allows them to properly route notifications and display relevant information to users without requiring additional synchronization requests.

In JMAP, an authenticated user may have access to multiple accounts, each containing distinct sets of mailboxes and messages. Each account is identified by a unique accountId, which serves as a namespace for all objects within that account. The accountId is a fundamental concept in JMAP architecture, as it allows servers to represent different data stores, organizational units, or delegation scenarios within a single authenticated session. For instance, a user might have access to their personal account, one or more shared team accounts, and delegated accounts from other users.

The JMAPACCESS extension, as currently specified in [RFC9698], implicitly assumes that all IMAP mailboxes accessible to an authenticated user correspond to a single JMAP account. Consequently, when object identifiers are exchanged between IMAP and JMAP, the accountId is omitted from the IMAP representation, as it is presumed to be constant. This assumption holds true in the majority of deployment scenarios, where a user's IMAP connection maps directly to their primary JMAP account.

However, this assumption breaks down in environments where an IMAP session provides access to mailboxes from multiple JMAP accounts. Such scenarios commonly arise when users access shared mailboxes, delegated folders, or organizational resources that exist within separate JMAP accounts but are presented as part of a unified IMAP mailbox hierarchy. In these cases, the MAILBOXID alone is insufficient to uniquely identify a mailbox across the IMAP-JMAP boundary, because different JMAP accounts may theoretically use the same MAILBOXID for different mailboxes.

This ambiguity becomes particularly problematic when considering the JMAP Email Delivery Push extension. When a client receives an EmailPush notification containing an accountId and a mailbox identifier, it needs to correlate this information with its IMAP

representation of the mailbox. Without an `accountId` in the IMAP context, the client cannot definitively determine which JMAP account a given IMAP mailbox belongs to, potentially leading to incorrect mailbox associations, failed synchronization attempts, or inability to properly handle push notifications for shared mailboxes.

The motivation for this specification is to extend the OBJECTID framework to include account-level identification in IMAP, thereby enabling precise correlation of mailboxes across IMAP and JMAP protocols in multi-account environments. By introducing the ACCOUNTID object identifier, this extension ensures that clients can unambiguously identify the JMAP account associated with each IMAP mailbox, facilitating interoperability and enabling advanced features such as account-aware push notifications.

### 1.1. Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

## 2. ACCOUNTID Object Identifier

The ACCOUNTID is a server-allocated identifier that specifies the JMAP account to which a mailbox belongs. When used in conjunction with the MAILBOXID, the ACCOUNTID provides complete disambiguation of mailboxes in environments where multiple JMAP accounts are accessible through a single IMAP session.

The ACCOUNTID is represented as an opaque string using the same character set and syntactic constraints as the MAILBOXID, as defined in Section 4 of [RFC8474]. That is, an ACCOUNTID MUST consist of between 1 and 255 characters from the base64url alphabet defined in Section 5 of [RFC4648], excluding the padding character.

The server MUST ensure that for any mailbox accessible via both IMAP and JMAP, the ACCOUNTID returned via IMAP matches the `accountId` property of the corresponding Mailbox object in JMAP, as defined in Section 1.6.2 of [RFC8620]. This correspondence is essential for clients to correlate mailboxes across the two protocols.

The server MUST return the same ACCOUNTID for all mailboxes that belong to the same JMAP account. Conversely, the server MUST NOT return the same ACCOUNTID for mailboxes that belong to different JMAP accounts, even if accessed within the same IMAP session.

When a mailbox is accessed exclusively through IMAP and does not have a corresponding representation in JMAP (for instance, if the server does not support the JMAPACCESS extension), the server MAY still assign an ACCOUNTID to maintain consistency in the IMAP representation. However, such ACCOUNTIDs need not correspond to any JMAP account identifier.

If a server advertises both "OBJECTID=ACCOUNTID" and "JMAPACCESS" capabilities, clients can rely on the ACCOUNTID to determine the appropriate JMAP account context for operations involving the mailbox. This is particularly important when handling JMAP Email Delivery Push notifications, as described in the introduction, where the accountId field in an EmailPush object can be correlated with the ACCOUNTID of the corresponding IMAP mailbox.

The ACCOUNTID is conceptually immutable for a given JMAP account within an IMAP session. However, if the underlying JMAP account is deleted or the user's access to that account is revoked, the associated mailboxes will no longer be accessible via IMAP, and their ACCOUNTIDs become irrelevant. If a new account is subsequently created with the same accountId, and mailboxes from that account become accessible via IMAP, the server MAY reuse the same ACCOUNTID value.

## 2.1. New Response Code for CREATE

This document extends the CREATE command to include the ACCOUNTID in the response code on successful mailbox creation, in addition to the MAILBOXID response code defined in Section 4.1 of [RFC8474].

A server advertising the "OBJECTID=ACCOUNTID" capability MUST include both the MAILBOXID and ACCOUNTID response codes in the tagged OK response to all successful CREATE commands.

Syntax: "ACCOUNTID" SP "(" objectid ")"

This is a response code in the tagged OK response for a successful CREATE command. The objectid uses the same syntax as defined for MAILBOXID in [RFC8474].

Example:

```
C: 3 create foo
S: 3 OK [MAILBOXID (F2212ea87-6097-4256-9d51-71338625) \
      ACCOUNTID (ula48e8e3)] Completed
C: 4 create shared/team
S: 4 OK [MAILBOXID (F8839dca12-3ef8-4a72-b63d-54f9e8a1) \
      ACCOUNTID (u2b59f9f4)] Completed
C: 5 create foo
S: 5 NO Mailbox already exists
```

In this example, the mailbox "foo" is created in account "ula48e8e3", while the shared mailbox "shared/team" is created in a different account "u2b59f9f4". This demonstrates how mailboxes within the same IMAP session can belong to different JMAP accounts.

## 2.2. New OK Untagged Response for SELECT and EXAMINE

This document adds the ACCOUNTID response code to the untagged OK responses returned by the SELECT and EXAMINE commands, complementing the MAILBOXID response code defined in Section 4.2 of [RFC8474].

A server advertising the "OBJECTID=ACCOUNTID" capability MUST return an untagged OK response with both the MAILBOXID and ACCOUNTID response codes on all successful SELECT and EXAMINE commands.

Syntax: "OK" SP "[" "ACCOUNTID" SP "(" objectid ")" "]" SP text

This is an untagged OK response to SELECT or EXAMINE commands.

Example:

```
C: 27 select "foo"
[...]
S: * OK [MAILBOXID (F2212ea87-6097-4256-9d51-71338625) \
      ACCOUNTID (ula48e8e3)] Ok
[...]
S: 27 OK [READ-WRITE] Completed

C: 28 select "shared/team"
[...]
S: * OK [MAILBOXID (F8839dca12-3ef8-4a72-b63d-54f9e8a1) \
      ACCOUNTID (u2b59f9f4)] Ok
[...]
S: 28 OK [READ-WRITE] Completed
```

This example shows how clients can determine the account context when selecting different mailboxes, even within the same IMAP session.

### 2.3. New Attribute for STATUS

This document adds the ACCOUNTID attribute to the STATUS command using the extended syntax defined in [RFC4466], in parallel with the MAILBOXID attribute defined in Section 4.3 of [RFC8474].

A server that advertises the "OBJECTID=ACCOUNTID" capability MUST support the ACCOUNTID status attribute.

Syntax: "ACCOUNTID"

This is the attribute name used in the STATUS command.

Syntax: "ACCOUNTID" SP "(" objectid ")"

This is the response item in the STATUS response, containing the account identifier assigned by the server for the mailbox.

Example:

```
C: 31 status foo (MAILBOXID ACCOUNTID MESSAGES)
S: * STATUS foo (MAILBOXID (F2212ea87-6097-4256-9d51-71338625)\
    ACCOUNTID (ula48e8e3) MESSAGES 42)
S: 31 OK Completed

C: 32 status shared/team (MAILBOXID ACCOUNTID MESSAGES)
S: * STATUS shared/team (\
    MAILBOXID (F8839dca12-3ef8-4a72-b63d-54f9e8a1)\
    ACCOUNTID (u2b59f9f4) MESSAGES 17)
S: 32 OK Completed

C: 33 list "" "*" return (status (MAILBOXID ACCOUNTID))
S: * LIST (\HasNoChildren) "." "foo"
S: * STATUS foo (MAILBOXID (F2212ea87-6097-4256-9d51-71338625)\
    ACCOUNTID (ula48e8e3))
S: * LIST (\HasNoChildren) "." "shared.team"
S: * STATUS shared.team (\
    MAILBOXID (F8839dca12-3ef8-4a72-b63d-54f9e8a1)\
    ACCOUNTID (u2b59f9f4))
S: 33 OK Completed
```

This example demonstrates how clients can efficiently retrieve both mailbox and account identifiers for multiple mailboxes using the extended LIST command with STATUS return option, as defined in [RFC5819].

### 3. Implementation Considerations

The implementation considerations specified in Section 8 of [RFC8474] apply equally to the ACCOUNTID object identifier defined in this specification.

In particular, the ACCOUNTID is generated and managed entirely by the server. Clients MUST NOT attempt to construct, modify, or interpret the internal structure of an ACCOUNTID value. The ACCOUNTID MUST be treated as an opaque identifier whose only meaningful operation is equality comparison.

Servers SHOULD generate ACCOUNTIDs using mechanisms that ensure global uniqueness across all accounts within the server's administrative domain, similar to the recommendations for MAILBOXID generation. Suitable approaches include Universally Unique Identifiers (UUIDs) as defined in [RFC4122], or server-assigned sequence numbers that are guaranteed never to be reused.

When both "OBJECTID=ACCOUNTID" and "JMAPACCESS" capabilities are advertised, servers MUST ensure that the ACCOUNTID values correspond exactly to the accountId values used in the JMAP protocol. This correspondence is essential for clients that need to correlate IMAP mailboxes with their JMAP representations, particularly when handling JMAP Email Delivery Push notifications.

Servers that provide access to mailboxes from multiple JMAP accounts through a single IMAP session SHOULD advertise the "OBJECTID=ACCOUNTID" capability to enable clients to distinguish between accounts. However, servers that provide access only to mailboxes from a single JMAP account MAY choose not to advertise this capability, as the account context is unambiguous in such cases.

Clients that support both IMAP and JMAP SHOULD use the ACCOUNTID when available to maintain accurate mappings between IMAP mailboxes and JMAP Mailbox objects. This is particularly important for clients that use JMAP Email Delivery Push notifications, as these notifications include the accountId property. By correlating the accountId from a push notification with the ACCOUNTID of IMAP mailboxes, clients can efficiently determine which IMAP mailbox corresponds to a newly delivered message without requiring additional synchronization operations.



Clients SHOULD be prepared to handle IMAP servers that do not advertise "OBJECTID=ACCOUNTID", even when the server advertises "JMAPACCESS". In such cases, clients MAY assume that all accessible mailboxes belong to a single JMAP account, though this assumption may lead to incorrect behavior if the server subsequently exposes mailboxes from additional accounts.

#### 4. Security Considerations

The security considerations specified in Section 11 of [RFC8474] and Section 7 of [RFC9698] apply to this specification.

The introduction of the ACCOUNTID object identifier does not fundamentally alter the security properties of the IMAP protocol or the OBJECTID extension. However, implementers should consider the following additional points.

##### 4.1. Account Identifier Exposure

The ACCOUNTID reveals information about the account structure of the server and which mailboxes belong to which accounts. While this information is generally not considered sensitive in the context of an authenticated IMAP session, servers that wish to minimize information disclosure MAY choose to generate ACCOUNTIDs using unpredictable values (such as UUIDs) rather than sequential numbers or other patterns that might reveal information about account creation order or the total number of accounts on the server.

##### 4.2. Cross-Account Information Leakage

Servers MUST ensure that the ACCOUNTID mechanism does not inadvertently grant users access to information about accounts they are not authorized to access. In particular, servers MUST NOT return ACCOUNTID values for accounts that the authenticated user does not have permission to access, even if such accounts exist on the server.

##### 4.3. Consistency with JMAP Authentication

When a server advertises both "OBJECTID=ACCOUNTID" and "JMAPACCESS" capabilities, the server MUST ensure that the same authentication credentials used for the IMAP session would grant access to the corresponding JMAP accounts. Inconsistencies in authentication or authorization between IMAP and JMAP could lead to situations where a client receives ACCOUNTIDs that it cannot subsequently use to access the corresponding JMAP resources, potentially revealing the existence of accounts the user cannot access.

#### 4.4. Digest-Based Identifier Generation

If servers use cryptographic digests to generate ACCOUNTIDs, the security considerations regarding collision resistance and algorithm selection discussed in Section 11 of [RFC8474] apply. Servers SHOULD monitor current security research and use digest algorithms that provide adequate collision resistance.

#### 4.5. Privacy in Multi-Tenant Environments

In multi-tenant or hosted environments, servers SHOULD generate ACCOUNTIDs in a manner that does not reveal relationships between accounts or organizational structures that users should not be aware of. For example, if multiple accounts belong to the same organization, the ACCOUNTID generation mechanism should not use patterns that would allow users to infer these relationships unless such information is explicitly intended to be visible.

#### 4.6. Other Considerations

See also the security considerations in [RFC3501], [RFC8620], and [RFC9051].

### 5. IANA Considerations

#### 5.1. IMAP Capabilities Registry

IANA is requested to add the following capability to the "Internet Message Access Protocol (IMAP) Capabilities Registry" located at <https://www.iana.org/assignments/imap-capabilities> (<https://www.iana.org/assignments/imap-capabilities>):

\*Capability name:\* OBJECTID=ACCOUNTID  
\*Reference:\* this document

#### 5.2. IMAP Response Codes Registry

IANA is requested to add the following response code to the "IMAP Response Codes" registry located at <https://www.iana.org/assignments/imap-response-codes> (<https://www.iana.org/assignments/imap-response-codes>):

\*Response code:\* ACCOUNTID  
\*Reference:\* this document

### 6. References

#### 6.1. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/rfc/rfc2119>>.
- [RFC4466] Melnikov, A. and C. Daboo, "Collected Extensions to IMAP4 ABNF", RFC 4466, DOI 10.17487/RFC4466, April 2006, <<https://www.rfc-editor.org/rfc/rfc4466>>.
- [RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/rfc/rfc8174>>.
- [RFC8474] Gondwana, B., Ed., "IMAP Extension for Object Identifiers", RFC 8474, DOI 10.17487/RFC8474, September 2018, <<https://www.rfc-editor.org/rfc/rfc8474>>.
- [RFC8620] Jenkins, N. and C. Newman, "The JSON Meta Application Protocol (JMAP)", RFC 8620, DOI 10.17487/RFC8620, July 2019, <<https://www.rfc-editor.org/rfc/rfc8620>>.
- [RFC9698] Gulbrandsen, A. and B. Gondwana, "The JMAPACCESS Extension for IMAP", RFC 9698, DOI 10.17487/RFC9698, January 2025, <<https://www.rfc-editor.org/rfc/rfc9698>>.

## 6.2. Informative References

- [I-D.ietf-jmap-emailpush] Jenkins, N., "JSON Meta Application Protocol (JMAP) Email Delivery Push Notifications", Work in Progress, Internet-Draft, draft-ietf-jmap-emailpush-01, 31 July 2025, <<https://datatracker.ietf.org/doc/html/draft-ietf-jmap-emailpush-01>>.
- [RFC3501] Crispin, M., "INTERNET MESSAGE ACCESS PROTOCOL - VERSION 4rev1", RFC 3501, DOI 10.17487/RFC3501, March 2003, <<https://www.rfc-editor.org/rfc/rfc3501>>.
- [RFC4122] Leach, P., Mealling, M., and R. Salz, "A Universally Unique IDentifier (UUID) URN Namespace", RFC 4122, DOI 10.17487/RFC4122, July 2005, <<https://www.rfc-editor.org/rfc/rfc4122>>.
- [RFC4648] Josefsson, S., "The Base16, Base32, and Base64 Data Encodings", RFC 4648, DOI 10.17487/RFC4648, October 2006, <<https://www.rfc-editor.org/rfc/rfc4648>>.

- [RFC5819] Melnikov, A. and T. Sirainen, "IMAP4 Extension for Returning STATUS Information in Extended LIST", RFC 5819, DOI 10.17487/RFC5819, March 2010, <<https://www.rfc-editor.org/rfc/rfc5819>>.
- [RFC9051] Melnikov, A., Ed. and B. Leiba, Ed., "Internet Message Access Protocol (IMAP) - Version 4rev2", RFC 9051, DOI 10.17487/RFC9051, August 2021, <<https://www.rfc-editor.org/rfc/rfc9051>>.

#### Appendix A. Changes

[[This section to be removed by RFC Editor]]

\*draft-degennaro-imap-objectid-accountid-00\*

\* Initial version

#### Author's Address

Mauro De Gennaro  
Stalwart Labs LLC  
1309 Coffeen Avenue, Suite 1200  
Sheridan, WY 82801  
United States of America  
Email: [mauro@stalw.art](mailto:mauro@stalw.art)  
URI: <https://stalw.art>