

```
MailViewerColumnMessageColor = ,eccf, \* Used to indicate sorting should be done by
attachments a message contains *\
```

```
MailViewerColumnAttachmentsColumn = ,ecaf, \* Column containing the number of
enum MailViewerColumn {
```

```
typedef enum MailSortingColor MailSortingColor;
};
```

```
MailSortingColorYellow = ,ccye, \* Yellow *\
MailSortingColorRed = ,ccre, \* Red *\
MailSortingColorPurple = ,ccbu, \* Purple *\
MailSortingColorOther = ,ccof, \* Other *\
MailSortingColorOrange = ,ccor, \* Orange *\
MailSortingColorGreen = ,ccgr, \* Green *\
MailSortingColorBlue = ,ccbf, \* Blue *\
```

```
enum MailSortingColor {
```

```
typedef enum MailDefaultMessageFormat MailDefaultMessageFormat;
};
```

```
MailDefaultMessageFormatRichFormat = ,qmrft, \* Rich Text *\
MailDefaultMessageFormatPlainFormat = ,qmprft, \* Plain Text *\
```

```
enum MailDefaultMessageFormat {
```

```
typedef enum MailSaveableFieldTypeFormat MailSaveableFieldTypeFormat;
};
```

```
MailSaveableFieldTypeNativeFormat = ,itfm, \* Native format *\
enum MailSaveableFieldTypeFormat {
```

```
typedef enum MailPrinttingErrorHandlerHandling MailPrinttingErrorHandlerHandling;
};
```

```
errors *\
```

```
MailPrinttingErrorHandlerHandlingDetailed = ,lwdt, \* print a detailed report of PostScript
MailPrinttingErrorHandlerHandlingStandard = ,lwdt, \* Standard PostScript error handling *\
```

```
enum MailPrinttingErrorHandlerHandling {
```

```
typedef enum MailSaveOptions MailSaveOptions;
};
```

```
MailSaveOptionsAsk = ,ask, \* Ask the user whether or not to save the file. *\
MailSaveOptionsNo = ,no, \* Do not save the file. *\
MailSaveOptionsYes = ,yes, \* Save the file. *\
```

```
enum MailSaveOptions {
```

```
MailCCRecipient, MailToRecipient, MailContainer, MailHeader, MailMailAttachment,
MailMailbox, MailRule, MailRuleCondition, MailRecipient, MailBCCRecipient,
MailAccount, MailIMAPAccount, MailICloudAccount, MailPOPAccount, MailSMTPServer,
MailLDAPServer, MailOLDMessageEditor, MailMessageViewer, MailSignature, MailMessage,
MailPassword, MailWord, MailCharacter, MailAttribute, MailOutgoingMessage,
@class MailApplication, MailDocument, MailWindow, MailRichText, MailAttachment,
```

```
#import <ScriptingBridge\ScriptingBridge.h>
```

```
#import <AppKit\AppKit.h>
```

```
*\
* Mail.h
\*
```

```
(qerrorcat). If you do, Mail will use the ,all messages but omit attachments, box *\  

MailMessageCriteriaBoxCriteriaHaveReceived = ,xmr, \* do not use this option  

box *\  

option (qerrorcat). If you do, Mail will use the ,all messages but omit attachments,  

MailMessageCriteriaBoxCriteriaNotKeptOnFileCriteriaHaveReceived = ,xmo, \* do not use this  

enum MailMessageCriteriaBoxCriteria {
```

```
typedef enum MailCriteriaCriteria MailCriteriaCriteria:  

}:
```

```
MailCriteriaCriteriaAll = ,ccl, \* all *\  

MailCriteriaCriteriaBox = ,ccl, \* box *\  

MailCriteriaCriteriaBoxCriteria = ,ccl, \* box criteria *\  

MailCriteriaCriteriaBoxCriteriaCriteria = ,ccl, \* box criteria criteria *\  

MailCriteriaCriteriaBoxCriteriaCriteriaCriteria = ,ccl, \* box criteria criteria criteria *\  

MailCriteriaCriteriaBoxCriteriaCriteriaCriteriaCriteria = ,ccl, \* box criteria criteria criteria criteria *\  

MailCriteriaCriteriaBoxCriteriaCriteriaCriteriaCriteriaCriteria = ,ccl, \* box criteria criteria criteria criteria criteria *\  

MailCriteriaCriteriaBoxCriteriaCriteriaCriteriaCriteriaCriteriaCriteria = ,ccl, \* box criteria criteria criteria criteria criteria criteria *\  

enum MailCriteriaCriteria {
```

```
typedef enum MailAuthentication MailAuthentication:  

}:
```

```
MailAuthenticationNone = ,ccl, \* none *\  

MailAuthenticationAppleToken = ,atok, \* Apple token *\  

certificates) *\  

MailAuthenticationExternal = ,ext, \* External authentication (TLS client  

MailAuthenticationMD2 = ,xmq, \* CFM-MD2 *\  

MailAuthenticationMIME = ,xmt, \* MIME *\  

MailAuthenticationKerberos = ,xk2, \* Kerberos V2 (GSSAPI) *\  

MailAuthenticationAPOP = ,apo, \* APOP *\  

MailAuthenticationPassword = ,xct, \* Clear text password *\  

enum MailAuthentication {
```

```
typedef enum MailViewColumns MailViewColumns:  

}:
```

```
message was saved *\  

MailViewColumnsDateAsReceivedColumn = ,crl, \* Column containing the date a draft  

*\  

MailViewColumnsToColumn = ,cto, \* Column containing the recipients of a message  

*\  

MailViewColumnsSubjectColumn = ,csu, \* Column containing the subject of a message  

MailViewColumnsSizeColumn = ,csz, \* Column containing the size of a message *\  

in a mailbox *\  

MailViewColumnsNumberColumn = ,cnu, \* Column containing the number of a message  

(read, unread, replied to, forwarded, etc) *\  

MailViewColumnsMessageStatusColumn = ,cms, \* Column indicating a messages status  

or account a message is in *\  

MailViewColumnsMailboxColumn = ,cmr, \* Column containing the name of the mailbox  

MailViewColumnsFromColumn = ,cfr, \* Column containing the sender's name *\  

MailViewColumnsFlagsColumn = ,cfl, \* Column containing the flags of a message *\  

sent *\  

MailViewColumnsDateSentColumn = ,cds, \* Column containing the date a message was  

was received *\  

MailViewColumnsDateReceivedColumn = ,cdr, \* Column containing the date a message  

col *
```

## @protocol MailSenderMethods

typedef enum MailTypeAccount MailTypeAccount;

{  
 MailTypeAccountUnknown = 'efuu', /\* Unknown \*/  
 MailTypeAccountICloud = 'efif', /\* iCloud \*/  
 MailTypeAccountiMac = 'efiw', /\* iMac \*/  
 MailTypeAccountZmail = 'efzw', /\* Zmail \*/  
 MailTypeAccountPop = 'efbo', /\* POP \*/  
enum MailTypeAccount {

typedef enum MailMessageType MailMessageType;

{  
 MailMessageTypeAttachment = 'faff', /\* Attachment type \*/  
 MailMessageTypeOtherHeader = 'ffoc', /\* To or CC header \*/  
 MailMessageTypeOther = 'ffoo', /\* To header \*/  
 MailMessageTypeSubjectHeader = 'fsup', /\* Subject header \*/  
 MailMessageTypeSenderInfo = 'fsid', /\* Sender is info \*/  
 MailMessageTypeSenderInfoNotMemberOfGroup = 'fsid', /\* Sender is not member of group \*/  
 MessageTypeRecipients /\*  
 MailMessageTypeSenderInfoNotInMailboxRecipients = 'fsup', /\* Sender is not in mailbox  
 MailMessageTypeSenderInfoNotInMailContacts = 'fsin', /\* Sender is not in mail contacts \*/  
 MailMessageTypeSenderInfoMemberOfGroup = 'fsiw', /\* Sender is member of group \*/  
 Recipients /\*  
 MailMessageTypeSenderInfoInMailboxRecipients = 'fsup', /\* Sender is in mailbox recipients  
 MailMessageTypeSenderInfoInMailContacts = 'fsin', /\* Sender is in mail contacts \*/  
 MailMessageTypeMessageIs Junk Mail = 'fwij', /\* Message is junk mail \*/  
 MailMessageTypeMessageContent = 'fwec', /\* Message content \*/  
 MailMessageTypeHeaderKey = 'fwqk', /\* An arbitrary header key \*/  
 MailMessageTypeFromHeader = 'firo', /\* From header \*/  
 MailMessageTypeAttachmentEventMessage = 'felw', /\* Event message \*/  
 MailMessageTypeCalendar = 'fccc', /\* CC header \*/  
 MailMessageTypeReplyRecipient = 'fsur', /\* Reply recipient \*/  
 MailMessageTypeAccount = 'facc', /\* Account \*/  
enum MailMessageType {

typedef enum MailMessageRelationship MailMessageRelationship;

{  
 MailMessageRelationshipNone = 'ldno', /\* Indicates no relationship is appropriate \*/  
 MailMessageRelationshipGreaterThanOrEqualTo = 'ldgt', /\* Greater than or equal to \*/  
 MailMessageRelationshipLessThan = 'ldlt', /\* Less than \*/  
 MailMessageRelationshipEqualTo = 'ldte', /\* Equal to \*/  
 MailMessageRelationshipEndsWith = 'ldew', /\* Ends with \*/  
 MailMessageRelationshipDoesNotContain = 'ldqn', /\* Does not contain \*/  
 MailMessageRelationshipDoesContain = 'ldco', /\* Does contain \*/  
 MailMessageRelationshipBeginningWith = 'ldpw', /\* Beginning with \*/  
enum MailMessageRelationship {

typedef enum MailMessageDisposition MailMessageDisposition;

{  
 MessageTypeAttachment /\*  
 MailMessageDispositionAllMessagesAndTheirAttachments = 'xajf', /\* All messages and  
 MessageTypeAttachment /\*  
 MailMessageDispositionAllMessagesButTheirAttachments = 'xapo', /\* All messages but

@interface MailDocument : NSObject <MailGenericsMethods>  
\\ A document.

@end

IMAP account with the server.

- (void) synchronizeWith:(MailAccount \*)with: \\ Command to trigger synchronizing of an of messages being acted upon.
- rules and menus that execute AppleScripts. The direct parameter of this method is a list (MailMailbox \*)inMailboxes forRule:(MailRule \*)forRule: \\ script method invoked by
- (void) beforeMailActionWithMessages:(NSArray<MailMessage \*> \*)x inMailboxes:
- (void) mailto:(NSString \*)x: \\ opens a mailto URL.
- (void) importMailMailboxAt:(NSURL \*)at: \\ imports a mailbox created by Mail.
- (void) cURL:(NSString \*)x: \\ opens a mailto URL.
- <ldoe@example.com>," as the direct object method return "john doe", as the direct parameter of a fully specified email address. E.g. calling this with "john doe
- (NSString \*) extractAddressFrom:(NSString \*)x: \\ Command to get just the email
- the direct object method return "john doe", as the direct parameter of a fully specified email address. E.g. calling this with "john doe <ldoe@example.com>," as
- (NSString \*) extractNameFrom:(NSString \*)x: \\ Command to get the full name out of a
- (void) checkForNewMailFor:(MailAccount \*)for: \\ triggers a check for email.
- (BOOL) exists:(id)x: \\ verify that an object exists.
- (void) delete:(MailMessage \*)message: \\ delete the application.
- (BOOL)delete:(id)x: \\ delete a document.
- (void) delete:(id)x withProperties:(NSDictionary \*)withProperties delete:(id)x:
- (id) open:(id)x: \\ open a document.

@property (copy, readonly) NSString \*version: \\ the version number of the application.

@property (copy, readonly) BOOL frontmost: \\ is this the active application?

@property (copy, readonly) NSString \*name: \\ the name of the application.

- (NSArray<MailWindow \*> \*) windows:
- (NSArray<MailDocument \*> \*) documents:

@interface MailApplication : NSObject

\\ the application's top-level scripting object.

\*\n\* 290914 2116\n/\*

@end

- (void) moveTo:(NSObject \*)to: \\ move an object to a new location.
- copy an object.
- (void) onScriptTo:(NSObject \*)to withProperties:(NSDictionary \*)withProperties: \\
- (void) delete: \\ delete an object.
- (BOOL)delete:(id)x: \\ delete a document.
- (void) deleteWithProperties:(NSDictionary \*)withProperties delete:(id)x:
- (void) save:(NSURL \*)in as:(MailSaveOptions \*)as: \\ save a document.
- document.
- (void) close:(MailMessage \*)message saveIn:(NSURL \*)saveIn: \\ close a

@blobelf (cobyl) ИЗУБГ \*titleName: \\ the title for the attachment

@interface MailAttachment : MailBichText

\\ Represents an inline text attachment. This class is used mainly for wake commands.

@end

@blobelf (cobyl) ИЗИМПЕР \*size: \\ the size in bytes of the first character.

@blobelf (cobyl) ИЗЗЛУНД \*font: \\ the name of the font of the first character.

@blobelf (cobyl) ИЗСОЛО \*color: \\ the color of the first character.

- (ЗВЕЖЕНУАЛЛАУ<MailAttachment \*> \*) attachments;
- (ЗВЕЖЕНУАЛЛАУ<MailAttribute \*> \*) attributes;
- (ЗВЕЖЕНУАЛЛАУ<MailCharacter \*> \*) characters;
- (ЗВЕЖЕНУАЛЛАУ<MailWord \*> \*) words;
- (ЗВЕЖЕНУАЛЛАУ<MailParagraph \*> \*) paragraphs;

@interface MailBichText : ZBObject <MailGenericMethod>

\\ Bich (style) text

\*\  
\* Text style  
\\*

@end

qizbjaleq in the window.

@blobelf (cobyl' leaqouyl) MailDocument \*document: \\ the document whose contents are

@blobelf B00Г zoomed: \\ Is the window zoomed right now?

@blobelf (leaqouyl) B00Г zoomed: \\ Does the window have a zoom button?

@blobelf B00Г list: \\ Is the window list right now?

@blobelf (leaqouyl) B00Г resized: \\ Can the window be resized?

@blobelf B00Г windowed: \\ Is the window windowed right now?

@blobelf (leaqouyl) B00Г windowed: \\ Does the window have a window button?

@blobelf (leaqouyl) B00Г closed: \\ Does the window have a close button?

@blobelf ИЗБЕСТ ponud: \\ the ponuding rectangle of the window.

@blobelf ИЗИТЕДЕР index: \\ the index of the window, ordered from top to back.

- (ИЗИТЕДЕР) id: \\ the unique identifier of the window.

@blobelf (cobyl' leaqouyl) ИЗЗЛУНД \*name: \\ the title of the window.

@interface MailWindow : ZBObject <MailGenericMethod>

\\ A window.

@end

@blobelf (cobyl' leaqouyl) ИЗУБГ \*title: \\ Its location on qizk, if it has one.

@blobelf (leaqouyl) B00Г windowed: \\ Has it been windowed since the last save?

@blobelf (cobyl' leaqouyl) ИЗЗЛУНД \*name: \\ Its name.

@blobelfy (cobyl) n2number \*size: \\ the size in bytes of the first character.  
@blobelfy (cobyl) n2string \*font: \\ the name of the font of the first character.  
@blobelfy (cobyl) n2color \*color: \\ the color of the first character.

- (ZBEJementArray<MailAttachment \*> \*) attachments:
- (ZBEJementArray<MailCharacter \*> \*) characters:
- (ZBEJementArray<MailWord \*> \*) words:
- (ZBEJementArray<MailParagraph \*> \*) paragraphs:

@interface MailAttribute : ZBObject <MailGenericMethod>  
\\ This subdivides the text into chunks that all have the same attributes.

@end

@blobelfy (cobyl) n2number \*size: \\ the size in bytes of the character.  
@blobelfy (cobyl) n2string \*font: \\ the name of the font of the character.  
@blobelfy (cobyl) n2color \*color: \\ the color of the character.

- (ZBEJementArray<MailAttachment \*> \*) attachments:
- (ZBEJementArray<MailAttribute \*> \*) attributes:

@interface MailCharacter : ZBObject <MailGenericMethod>  
\\ This subdivides the text into characters.

@end

@blobelfy (cobyl) n2number \*size: \\ the size in bytes of the first character.  
@blobelfy (cobyl) n2string \*font: \\ the name of the font of the first character.  
@blobelfy (cobyl) n2color \*color: \\ the color of the first character.

- (ZBEJementArray<MailAttachment \*> \*) attachments:
- (ZBEJementArray<MailAttribute \*> \*) attributes:
- (ZBEJementArray<MailCharacter \*> \*) characters:

@interface MailWord : ZBObject <MailGenericMethod>  
\\ This subdivides the text into words.

@end

@blobelfy (cobyl) n2number \*size: \\ the size in bytes of the first character.  
@blobelfy (cobyl) n2string \*font: \\ the name of the font of the first character.  
@blobelfy (cobyl) n2color \*color: \\ the color of the first character.

- (ZBEJementArray<MailAttachment \*> \*) attachments:
- (ZBEJementArray<MailAttribute \*> \*) attributes:
- (ZBEJementArray<MailCharacter \*> \*) characters:
- (ZBEJementArray<MailWord \*> \*) words:

@interface MailParagraph : ZBObject <MailGenericMethod>  
\\ This subdivides the text into paragraphs.

@end

@briberfly (readonly) n2inteder packdroonqactivitlycount: \\ number of packdroonq  
 tetches of new mail, -J means to use an automatically determined interval  
 @briberfly n2inteder tetchinterval: \\ the interval (in minutes) between automatic  
 sbbjtsctiou  
 @briberfly (cobyl, readonly) n22trind \*sbbjtsctiouversion: \\ the putiq number of the  
 the nser pas zefecteq  
 @briberfly (cobyl, readonly) n2Arlay<mailmessage \*> \*selection: \\ list of messages that  
 ttejq of messages mptch lon are combosind  
 @briberfly B00G 9jmay2ccmlysejt: \\ Indiqates mptcher lon mtlj pe tucjndeq in the cc:  
 ttejq of messages mptch lon are combosind  
 @briberfly B00G 9jmay2bccmlysejt: \\ Indiqates mptcher lon mtlj pe tucjndeq in the bcc:

- (2BEJewentArlay<mail2idnagture \*> \*) 2idnagture:
- (2BEJewentArlay<mailBnje \*> \*) lnjes:
- (2BEJewentArlay<mailmessageVtemel \*> \*) messageVtemels:
- (2BEJewentArlay<mailmailprox \*> \*) mailproxes:
- (2BEJewentArlay<mailoutgoindmessage \*> \*) outgoindmessages:
- (2BEJewentArlay<mail2wfb2elvel \*> \*) 2wfb2elvels:
- (2BEJewentArlay<mailICJondqAccount \*> \*) ICJondqAccounts:
- (2BEJewentArlay<mailImagbAccount \*> \*) ImagbAccounts:
- (2BEJewentArlay<mailBobAccount \*> \*) bobAccounts:
- (2BEJewentArlay<mailAccount \*> \*) accounts:

@interfacc MailVbbsjtsctiou (mail)  
 \\ mail's tob levej 2cljbtind opject.

@enuq

- (B00G) 2enuq: \\ 2enuq 9 message.
- (n2inteder) tq: \\ the nuidne identifier of the message
- @briberfly (cobyl) tq message2idnagture: \\ the 2idnagture of the message  
 the defaulj is false
- @briberfly B00G v22trpe: \\ controls mptcher the message window is shown on the screen.
- @briberfly (cobyl) mailBtchtext \*content: \\ the contents of the message
- @briberfly (cobyl) n22trind \*2upject: \\ the 2upject of the message
- @briberfly (cobyl) n22trind \*2ender: \\ the 2ender of the message

- (2BEJewentArlay<mailtoBectbient \*> \*) toBectbients:
- (2BEJewentArlay<mailBectbient \*> \*) lectbients:
- (2BEJewentArlay<mailccBectbient \*> \*) ccBectbients:
- (2BEJewentArlay<mailbccBectbient \*> \*) bccBectbients:

@interfacc MailOutgoindMessage : 2BObject <MailGenerictMethods>  
 \\ A new email message

```
*\
* Mail
\*
```

@enuq

[illegible]



- (ZBEJementArray<MailCCRecipient \*> \*) ccrecipients;
- (ZBEJementArray<MailBCCRecipient \*> \*) bccrecipients;

@interface MailMessage : ZBObject <MailGenericMethods>  
 \\ An email message

\*\  
 \* Mail Framework  
 \\*

@end

@property (copy) NSString \*name; \\ Name of the signature  
 version with font and/or styles, that will be returned over the plain text version  
 @property (copy) NSString \*content; \\ Contents of email signature. If there is a

@interface MailSignature : ZBObject <MailGenericMethods>  
 \\ Email signatures

@end

@property (copy) MailWindow \*window; \\ The window for the message viewer  
 currently selected in the list of mailboxes  
 @property (copy) NSArray<MailMailbox \*> \*selectedMailboxes; \\ List of mailboxes  
 selected  
 @property (copy) NSArray<MailMessage \*> \*selectedMessages; \\ List of messages currently  
 being displayed in the viewer  
 @property (copy) NSArray<MailMessage \*> \*visibleMessages; \\ List of messages currently  
 - (NSInteger) id; \\ The unique identifier of the message viewer  
 that are visible. The subject column and the message status column will always be visible  
 @property (copy) NSArray<ZABBJEventDescriptor \*> \*visibleColumns; \\ List of columns  
 viewer window is visible or not  
 @property BOOL brevityEnabledVisible; \\ Controls whether the brevity bar of the message  
 or not  
 @property BOOL mailboxListVisible; \\ Controls whether the list of mailboxes is visible  
 @property BOOL sortedAscending; \\ Whether the viewer is sorted ascending or not  
 viewer  
 @property MailViewerColumns sortColumn; \\ The column that is currently sorted in the  
 @property (copy, readonly) MailMailbox \*trashMailbox; \\ The top level trash mailbox  
 @property (copy, readonly) MailMailbox \*sentMailbox; \\ The top level sent mailbox  
 @property (copy, readonly) MailMailbox \*inbox; \\ The top level out mailbox  
 @property (copy, readonly) MailMailbox \*junkMailbox; \\ The top level junk mailbox  
 @property (copy, readonly) MailMailbox \*inbox; \\ The top level in mailbox  
 @property (copy, readonly) MailMailbox \*draftsMailbox; \\ The top level drafts mailbox

- (ZBEJementArray<MailMessage \*> \*) messages;

@interface MailMessageViewer : ZBObject <MailGenericMethods>  
 \\ Represents the object responsible for managing a viewer window

@end

@broberty MailAuthentication authentication: \\ Preferred authentication scheme for read via scripting

@broberty (cobyl) Message \*password: \\ Password for this account. Can be set, but not - (Message \*) id: \\ The unique identifier of the account

@broberty (cobyl) Message \*name: \\ The name of an account this account

@broberty (cobyl) id deliveryaccount: \\ The delivery account used when sending mail from

- (SBELEMENTARRAY<MailMailbox \*> \*) mailboxes:

Interlace MailAccount : Object <MailGenericMethods>  
 use the 'bob account', 'jwab account', and 'jclonq account' objects  
 \\ A Mail account for receiving messages (BOB\IMAB). To create a new receiving account'

Send

(BOOG)replytoall: \\ Creates a reply message.  
 - (MailJonathanMessage \*) replyJonathanWindow:(BOOG)JonathanWindow replytoall:  
 redirected message.  
 - (MailJonathanMessage \*) redirectJonathanWindow:(BOOG)JonathanWindow: \\ Creates a  
 forwarded message.  
 - (MailJonathanMessage \*) forwardJonathanWindow:(BOOG)JonathanWindow: \\ Creates a

or not

@broberty (readonly) BOOG wasRepliedto: \\ Indicates whether the message was replied to  
 or not

@broberty (readonly) BOOG wasRedirected: \\ Indicates whether the message was redirected  
 or not

@broberty (readonly) BOOG wasForwarded: \\ Indicates whether the message was forwarded

@broberty (cobyl, readonly) Message \*subject: \\ The subject of the message

@broberty (cobyl, readonly) Message \*sender: \\ The sender of the message

@broberty (readonly) MessageSize: \\ The size (in bytes) of a message  
 to

@broberty (cobyl, readonly) Message \*replyto: \\ The address that replies should be sent

@broberty (cobyl, readonly) Message \*source: \\ Raw source of the message

@broberty (cobyl, readonly) Message \*messageid: \\ The unique message ID string

@broberty BOOG readstatus: \\ Indicates whether the message is read or not  
 evaluated to be true by the true mail filter.

@broberty BOOG JunkMailStatus: \\ Indicates whether the message has been marked junk or  
 flagged

@broberty MessageFlagIndex: \\ The flag on the message, or -1 if the message is not

@broberty BOOG flaggedstatus: \\ Indicates whether the message is flagged or not

@broberty BOOG deletedstatus: \\ Indicates whether the message is deleted or not

@broberty (cobyl, readonly) Message \*dateSent: \\ The date a message was sent

@broberty (cobyl, readonly) Message \*dateReceived: \\ The date a message was received

@broberty (cobyl, readonly) MailBodyText \*content: \\ Contents of an email message

@broberty (cobyl) MailMailbox \*mailbox: \\ The mailbox in which this message is filed

@broberty MailHeaderColorPackColor: \\ The packcolor of the message

@broberty (cobyl, readonly) Message \*headers: \\ All the headers of the message  
 - (Message) id: \\ The unique identifier of the message.

- (SBELEMENTARRAY<MailMailAttachment \*> \*) mailAttachments:  
 - (SBELEMENTARRAY<MailHeader \*> \*) headers:  
 - (SBELEMENTARRAY<MailToRecipient \*> \*) toRecipients:  
 - (SBELEMENTARRAY<MailRecipient \*> \*) recipients:

@property NSInteger messageIdMarkerSize: \\ If message size (in bytes) is over this

interface MailboxAccount : MailAccount

\\ A POP email account

@end

interface MailICloudAccount : MailIMAPAccount

\\ An iCloud or MobileMe email account

@end

be stored on the IMAP server

@property BOOL storeDeletedMessagesOnServer: \\ Indicates whether deleted messages will  
stored on the IMAP server

@property BOOL storeSentMessagesOnServer: \\ Indicates whether sent messages will be  
the IMAP server

@property BOOL store JunkMailOnServer: \\ Indicates whether junk mail will be stored on  
IMAP server

@property BOOL storeDraftsOnServer: \\ Indicates whether drafts will be stored on the  
account

@property MailMessageCachingPolicy messageCaching: \\ Message caching setting for this  
automatically compacted when you draft mail or switch to another mailbox

@property BOOL compactMailboxesWhenClosing: \\ Indicates whether an IMAP mailbox is

interface MailIMAPAccount : MailAccount

\\ An IMAP email account

@end

@property BOOL purge: \\ Indicates whether purge is enabled for this receiving account  
deleted will be moved to the trash mailbox

@property BOOL moveDeletedMessagesToTrash: \\ Indicates whether messages that are

@property (copy) NSString \*serverName: \\ The host name used to connect to an account

@property NSInteger port: \\ The port used to connect to an account

stores things on disk

@property (copy, readonly) NSString \*accountDirectory: \\ The directory where the account

@property (copy) NSString \*userName: \\ The user name used to connect to an account

@property BOOL enabled: \\ Indicates whether the account is enabled or not

mailboxes will be permanently deleted on disk

@property BOOL emptyTrashOnDisk: \\ Indicates whether the messages in deleted messages  
messages mailboxes will be deleted on disk

@property BOOL emptyJunkMessagesOnDisk: \\ Indicates whether the messages in the junk  
are permanently deleted (0 = delete on disk, -1 = never delete)

@property NSInteger emptyTrashFrequency: \\ Number of days before messages in the trash  
are deleted (0 = delete on disk, -1 = never delete)

@property NSInteger emptyJunkMessagesFrequency: \\ Number of days before junk messages

@property (copy) NSString \*fullName: \\ The user's full name configured for an account  
configured for an account

@property (copy) NSArray<NSString\*> \*emailAddresses: \\ The list of email addresses

@property (readonly) MailTypeOfAccount accountType: \\ The type of an account  
account

forwarded message. Set to empty string to include no forwarded text  
@memberof (cobyl) returning \*forwardText: \\ It will match, being this text to the  
@memberof BOG deleteMessage: \\ It will match, delete message  
@memberof MailHeaderColor colorMessage: \\ It will match, apply this color

- (ZBElementArray<MailRuleCondition \*> \*) ruleConditions:

@interface MailRule : ZBObject <MailGenericMethods>

\\ Class for message rules

@end

@memberof (cobyl, readonly) MailMailbox \*container:

@memberof (cobyl, readonly) MailAccount \*account:

mailbox

@memberof (readonly) NSInteger unreadCount: \\ The number of unread messages in the

@memberof (cobyl) returning \*name: \\ The name of a mailbox

- (ZBElementArray<MailMessage \*> \*) messages:

- (ZBElementArray<MailMailbox \*> \*) mailboxes:

@interface MailMailbox : ZBObject <MailGenericMethods>

\\ A mailbox that holds messages

@end

@memberof BOG pageSize: \\ Indicates whether ZL is enabled for this receiving account

@memberof (cobyl) returning \*serverName: \\ The host name used to connect to an account

@memberof NSInteger port: \\ The port used to connect to an account

@memberof (cobyl) returning \*userName: \\ The user name used to connect to an account

@memberof BOG enabled: \\ Indicates whether the account is enabled or not

account

@memberof MailAuthentication authentication: \\ Preferred authentication scheme for

@memberof (readonly) MailTypedAccount accountType: \\ The type of an account

read via scripting

@memberof (cobyl) returning \*password: \\ Password for this account. Can be set, but not

@memberof (cobyl, readonly) returning \*name: \\ The name of an account

@interface MailSMTPServer : ZBObject <MailGenericMethods>

\\ An SMTP account (for sending email)

@end

deleted from the server when moved from your POP inbox

@memberof BOG deleteMessagesWhenMovedFromInbox: \\ Indicates whether messages will be  
the server after downloading

@memberof BOG deleteMailFromServer: \\ Indicates whether POP account deletes messages on  
downloading)

that have been downloaded will be deleted from the server (0 = delete immediately after

@memberof NSInteger deleteMessageDeletionInterval: \\ Number of days before messages  
browbe)

account, Mail will browbe you asking whether you want to download the message (-1 = do not

@interface MailBccRecipient : MailRecipient  
\\ An email recipient in the BCC: field

@end

@property (copy) NSString \*name: \\ The name used for display  
@property (copy) NSString \*address: \\ The recipient's email address

@interface MailRecipient : NSObject <MailGenericMethod>  
\\ An email recipient

@end

@property MailButtonType type: \\ Button type  
@property MailButtonStyleType style: \\ Button style  
@property (copy) NSString \*label: \\ Button label key  
@property (copy) NSString \*expression: \\ Button expression field

@interface MailButtonCondition : NSObject <MailGenericMethod>  
\\ Class for conditions that can be attached to a button type

@end

message

@property BOOL shouldEvaluate: \\ If type matches, should evaluate for this  
@property BOOL shouldMoveMessage: \\ Indicates whether the type has a move action  
@property BOOL shouldCopyMessage: \\ Indicates whether the type has a copy action  
@property (copy) NSString \*name: \\ Name of type  
@property BOOL enabled: \\ Indicates whether the type is enabled  
mutually the text or password of a message in the message list  
@property BOOL shouldHighlightColor: \\ Indicates whether the color will be used to  
@property (copy) Mailbox \*mailbox: \\ If type matches, move to this mailbox  
copy to this mailbox  
@property (copy) Mailbox \*copyMessage NOT\_RETAINED: \\ If type matches,  
for type to execute  
@property BOOL shouldNotify: \\ Indicates whether all conditions must be met  
set to empty string to disable this action  
@property (copy) id userInfo: \\ If type matches, use this combined AppleScript file  
with this text. set to empty string to disable this action  
@property (copy) NSString \*replyText: \\ If type matches, reply to message and bubble  
action  
address or multiple addresses, separate by comma. set to empty string to disable this  
@property (copy) NSString \*redirectMessage: \\ If type matches, redirect message to this  
of group or list to group)  
@property (copy) NSString \*displayName: \\ If type matches, display this group (specific name  
@property BOOL markRead: \\ If type matches, mark message as read  
flag. set to -1 to disable this action  
@property NSInteger markFlagIndex: \\ If type matches, mark message with the specified  
@property BOOL markFlagged: \\ If type matches, mark message as flagged  
action  
address, or multiple addresses, separate by comma. set to empty string to disable this  
@property (copy) NSString \*forwardMessage: \\ If type matches, forward message to this

@end

- (NSString \*) id: \\ The unique identifier of the attachment.  
qomusodseq.

@property (readonly) BOOL qomusodseq: \\ Indicates whether the attachment has been

@property (readonly) NSInteger filesize: \\ Approximate size in bytes.

byayn.

@property (copy, readonly) NSString \*MIMEType: \\ MIME type of the attachment E.g. text\

@property (copy, readonly) NSString \*name: \\ Name of the attachment

@interface MailMailAttachment : NSObject <MailGenericMethods>

\\ A file attached to a received message.

@end

@property (copy) NSString \*name: \\ Name of the header value

@property (copy) NSString \*content: \\ Contents of the header

@interface MailHeader : NSObject <MailGenericMethods>

\\ A header value for a message. E.g. To, Subject, From.

@end

@interface MailContainer : MailMailbox

\\ A mailbox that contains other mailboxes.

@end

@interface MailToRecipient : MailRecipient

\\ An email recipient in the To: field

@end

@interface MailCCRecipient : MailRecipient

\\ An email recipient in the CC: field

@end