

AS2Secure Php Lib Documentation



Contents

Package Horde_Horde_Util_Procedural_Elements	2
Util.php	2
Define HORDE_ERROR_DRIVER_CONFIG	2
Define HORDE_ERROR_DRIVER_CONFIG_MISSING	2
Package Horde_Horde_Util_Classes	3
Class Horde_Util	3
Var \$cache	3
Var \$files	4
Var \$securedel	4
Method cloneObject	4
Method deleteAtShutdown	4
Method extensionExists	5
Method getTempFile	5
Method deleteAtShutdown	6
Package Horde_MIME_Procedural_Elements	8
MIME.php	8
Define ENC7BIT	8
Define ENC8BIT	8
Define ENCBASE64	8
Define ENCBINARY	8
Define ENCOTHER	8
Define ENCQUOTEDPRINTABLE	8
Define MIME_CODE	8
Define MIME_STRING	8
Define TYPEAPPLICATION	8
Define TYPEAUDIO	8
Define TYPEIMAGE	8
Define TYPMESSAGE	8
Define TYPMODEL	8
Define TYPMULTIPART	8
Define TYPEOTHER	8
Define TYPETEXT	8
Define TYPEVIDEO	8
Part.php	9
Define MIME_DEFAULT_CHARSET	9
Define MIME_DEFAULT_DESCRIPTION	9
Define MIME_DEFAULT_DISPOSITION	9
Define MIME_DEFAULT_ENCODING	9
Define MIME_PART_EOL	9
Define MIME_PART_RFC_EOL	9
Package Horde_MIME_Classes	10
Class Horde_MIME	10

Var \$mime_encodings	10
Var \$mime_types	11
Var \$rfc822_filter	11
Method addrArray2String	11
Method addrObject2String	12
Method bareAddress	12
Method decode	13
Method decodeAddrString	13
Method decodeRFC2231	13
Method encode	14
Method encodeAddress	14
Method encodeRFC2231	15
Method encoding	15
Method generateMessageID	15
Method is8bit	16
Method parseAddressList	16
Method quotedPrintableEncode	17
Method rfc822Explode	17
Method rfc822WriteAddress	17
Method trimEmailAddress	18
Method type	18
Method wrapHeaders	18
Method encode	19
Method getCode	19
Method rfc822Encode	20
Class Horde_MIME_Message	20
Var \$ build	21
Var \$ defaultServer	21
Constructor construct	21
Method buildMessage	21
Method convertMIMEPart	22
Method encode	22
Method getBasePart	22
Method getPart	23
Method getParts	23
Method header	23
Method toString	24
Class Horde_MIME_Part	24
Var \$ boundary	24
Var \$ bytes	25
Var \$ cids	25
Var \$ contentid	25
Var \$ contents	26
Var \$ contentTypeParameters	26
Var \$ currentEncoding	26
Var \$ description	26
Var \$ disposition	27
Var \$ dispositionParameters	27
Var \$ encode7bit	27

Var \$ eol	27
Var \$ flags	28
Var \$ idmap	28
Var \$ information	28
Var \$ mimeid	29
Var \$ parts	29
Var \$ subtype	29
Var \$ transferEncoding	29
Var \$ type	30
Constructor <code>construct</code>	30
Method <code>addCID</code>	30
Method <code>addPart</code>	31
Method <code>alterPart</code>	31
Method <code>appendContents</code>	31
Method <code>clearContents</code>	32
Method <code>clearContentTypeParameter</code>	32
Method <code>contentTypeMap</code>	32
Method <code>getAllContentTypeParameters</code>	33
Method <code>getAllDispositionParameters</code>	33
Method <code>getBytes</code>	33
Method <code>getCanonicalContents</code>	33
Method <code>getCharset</code>	34
Method <code>getCIDList</code>	34
Method <code>getContentID</code>	34
Method <code>getContents</code>	34
Method <code>getContentTypeParameter</code>	34
Method <code>getCurrentEncoding</code>	35
Method <code>getDefaultType</code>	35
Method <code>getDescription</code>	35
Method <code>getDisposition</code>	36
Method <code>getDispositionParameter</code>	36
Method <code>getEOL</code>	36
Method <code>getInformation</code>	36
Method <code>getMIMEId</code>	37
Method <code>getName</code>	37
Method <code>getPart</code>	37
Method <code>getParts</code>	38
Method <code>getPrimaryType</code>	38
Method <code>getRelativeMIMEId</code>	38
Method <code>getSubType</code>	39
Method <code>getTransferEncoding</code>	39
Method <code>getType</code>	39
Method <code>getUniqueID</code>	39
Method <code>header</code>	40
Method <code>removePart</code>	40
Method <code>replaceEOL</code>	40
Method <code>setBytes</code>	41
Method <code>setCharset</code>	41
Method <code>setContentID</code>	41

Method setContents	42
Method setContentTypeParameter	42
Method setDescription	43
Method setDisposition	43
Method setDispositionParameter	43
Method setEOL	44
Method setInformation	44
Method setMIMEId	44
Method setName	45
Method setTransferEncoding	45
Method setType	45
Method splitContents	46
Method strict7bit	46
Method toCanonicalString	46
Method toString	47
Method transferDecode	47
Method transferDecodeContents	47
Method transferEncode	48
Method transferEncodeContents	48
Method generateBoundary	48
Method generateIdMap	48
Method partFind	49
Class Horde MIME Structure	49
Method addMultipartInfo	50
Method parse	50
Method parseMIMEHeaders	50
Method parseTextMIMEMessage	51
Method convertMimeDecodeData	51
Method getParameters	51
Method parse	52
Method setInfo	52
Package AS2Secure Classes	54
Class AS2Abstract	54
Var \$adapter	54
Var \$filename	55
Var \$files	55
Var \$headers	55
Var \$is_crypted	55
Var \$is_signed	55
Var \$message_id	56
Var \$mimetype	56
Var \$partner_from	56
Var \$partner_to	56
Var \$path	56
Constructor __construct	57
Method addFile	57
Method decode	57
Method encode	57
Method generateMessageID	58

Method getAuthentication	58
Method getContent	58
Method getFileName	58
Method getFiles	58
Method getHeader	58
Method getHeaders	59
Method getMessageId	59
Method getPartnerFrom	59
Method getPartnerTo	59
Method getPath	59
Method getUrl	59
Method isCrypted	60
Method isSigned	60
Method setHeaders	60
Method setMessageId	60
Method setPartnerFrom	61
Method setPartnerTo	61
Class AS2Adapter	61
Var \$javapath	62
Var \$partner_from	62
Var \$partner_to	62
Var \$ssl_adapter	62
Var \$ssl_openssl	63
Var \$tmp_files	63
Constructor __construct	63
Method addTempFileForDelete	64
Method calculateMicChecksum	64
Method compose	64
Method compress	65
Method decompress	65
Method decrypt	65
Method detectMimeType	66
Method detectOS	66
Method encrypt	66
Method exec	67
Method extract	67
Method getCAFromPKCS12	67
Method getDataFromPKCS12	68
Method getMicChecksum	68
Method getPrivateFromPKCS12	68
Method getPublicFromPKCS12	69
Method getTempFilename	69
Method hex2bin	69
Method sign	70
Method verify	70
Method deleteTempFiles	71
Class AS2Client	71
Var \$response_content	72
Var \$response_headers	72

Var \$response indice	72
Constructor construct	72
Method getLastResponse	72
Method handleResponseHeader	73
Method sendRequest	73
Class AS2Connector	73
Class Constant STATUS_ERROR	74
Class Constant STATUS_OK	74
Method onReceivedMDN	74
Method onReceivedMessage	74
Method onSentMDN	75
Method onSentMessage	75
Class AS2Exception	76
Class Constant DEFAULT_ERROR	76
Class Constant STATUS_ERROR	76
Class Constant STATUS_FAILURE	76
Class Constant STATUS_WARNING	76
Var \$level_error	77
Var \$level_failure	77
Var \$level_warning	77
Constructor construct	77
Method getLevel	78
Method getMessageShort	78
Class AS2Header	78
Var \$headers	79
Var \$ position	79
Constructor construct	79
Method addHeader	79
Method addHeaders	80
Method addHeadersFromMessage	80
Method count	81
Method current	81
Method exists	81
Method getHeader	81
Method getHeaders	82
Method key	82
Method next	82
Method offsetExists	82
Method offsetGet	82
Method offsetSet	83
Method offsetUnset	83
Method parseHttpRequest	83
Method parseText	84
Method removeHeader	84
Method rewind	84
Method setHeaders	85
Method toFormattedArray	85
Method valid	85
Method toString	85

Class AS2Log	86
Class Constant ERROR	86
Class Constant FAILURE	86
Class Constant INFO	86
Class Constant WARNING	86
Var \$code	86
Var \$current_message_id	87
Var \$filename	87
Var \$level	87
Var \$message	87
Var \$message_id	88
Var \$stack	88
Constructor construct	88
Method error	88
Method getCode	89
Method getCount	89
Method getLastLogEvents	89
Method getLevel	90
Method getMessage	90
Method getMessageId	90
Method getStack	90
Method handleLevel	91
Method hasError	91
Method info	91
Method logEvent	92
Method warning	92
Class AS2MDN	93
Class Constant ACTION_AUTO	93
Class Constant ACTION_MANUAL	93
Class Constant MODIFIER_ERROR	93
Class Constant MODIFIER_WARNING	93
Class Constant SENDING_AUTO	94
Class Constant SENDING_MANUAL	94
Class Constant TYPE_FAILED	94
Class Constant TYPE_PROCESSED	94
Var \$attributes	94
Var \$message	94
Var \$url	95
Constructor construct	95
Method decode	95
Method encode	95
Method getAttribute	96
Method getAttributes	96
Method getMessage	96
Method getUrl	96
Method setAttribute	97
Method setMessage	97
Method toString	97
Class AS2Message	98

Var \$mic_checksum	98
Constructor construct	98
Method addFile	99
Method decode	99
Method encode	99
Method generateMDN	100
Method getAuthentication	100
Method getFiles	100
Method getMicChecksum	100
Method getUrl	101
Class AS2Partner	101
Class Constant ACK_ASYNC	101
Class Constant ACK_SYNC	102
Class Constant CRYPT_3DES	102
Class Constant CRYPT_AES_128	102
Class Constant CRYPT_AES_192	102
Class Constant CRYPT_AES_256	102
Class Constant CRYPT_DES	102
Class Constant CRYPT_NONE	102
Class Constant CRYPT_RC2_40	102
Class Constant CRYPT_RC2_64	102
Class Constant CRYPT_RC2_128	102
Class Constant ENCODING_BASE64	102
Class Constant ENCODING_BINARY	102
Class Constant METHOD_AUTO	102
Class Constant METHOD_BASIC	102
Class Constant METHOD_DIGEST	102
Class Constant METHOD_GSS	103
Class Constant METHOD_NONE	103
Class Constant METHOD_NTLM	103
Class Constant SIGN_MD5	103
Class Constant SIGN_NONE	103
Class Constant SIGN_SHA1	103
Var \$comment	103
Var \$connector_class	103
Var \$email	103
Var \$id	104
Var \$is_local	104
Var \$mdn_credencial_login	104
Var \$mdn_credencial_method	104
Var \$mdn_credencial_password	104
Var \$mdn_request	105
Var \$mdn_signed	105
Var \$mdn_subject	105
Var \$mdn_url	105
Var \$name	105
Var \$sec_certificate	106
Var \$sec_encrypt_algorithm	106
Var \$sec_pkcs12	106

Var \$sec_pkcs12_password	106
Var \$sec_signature_algorithm	106
Var \$send_compress	107
Var \$send_content_type	107
Var \$send_credencial_login	107
Var \$send_credencial_method	107
Var \$send_credencial_password	107
Var \$send_encoding	108
Var \$send_subject	108
Var \$send_url	108
Var \$stack	108
Constructor <u>construct</u>	108
Method <u>getAvailablesEncryptions</u>	109
Method <u>getAvailablesSignatures</u>	109
Method <u>getPartner</u>	109
Method <u>get</u>	110
Method <u>set</u>	110
Method <u>toString</u>	110
Class AS2Request	111
Var \$request	111
Constructor <u>construct</u>	111
Method <u>decode</u>	112
Method <u>decrypt</u>	112
Method <u>encode</u>	112
Method <u>getObject</u>	112
Class AS2Server	112
Class Constant <u>TYPE_MDN</u>	113
Class Constant <u>TYPE_MESSAGE</u>	113
Method <u>closeConnectionAndWait</u>	113
Method <u>handle</u>	114
Method <u>saveMessage</u>	114
Package Horde_Util Classes	115
Class Horde_String	115
Var \$charset	115
Method <u>convertCharset</u>	115
Method <u>extensionExists</u>	116
Method <u>isAlpha</u>	116
Method <u>isLower</u>	117
Method <u>isUpper</u>	117
Method <u>length</u>	117
Method <u>lower</u>	118
Method <u>pad</u>	118
Method <u>pos</u>	119
Method <u>regexMatch</u>	119
Method <u>setDefaultCharset</u>	120
Method <u>substr</u>	120
Method <u>ucfirst</u>	120
Method <u>upper</u>	121
Method <u>wordwrap</u>	121

Method wrap	122
Method convertCharset	122
Method mbstringCharset	123
Appendices	124
Appendix A - Class Trees	125
Horde MIME	125
AS2Secure	125
Horde Util	127
Horde Horde Util	127

Package Horde_Horde_Util Procedural Elements

Util.php

- **Package** Horde_Horde_Util

HORDE_ERROR_DRIVER_CONFIG = 2 [*line 11*]

Error code for an incomplete driver configuration.

HORDE_ERROR_DRIVER_CONFIG_MISSING = 1 [*line 6*]

Error code for a missing driver configuration.

Package Horde_Horde_Util Classes

Class Horde_Util

[line 28]

The Horde_Util:: class provides generally useful methods of different kinds.

\$Horde: framework/Horde_Util/Horde_Util.php,v 1.384.6.37 2009/07/21 18:17:23 slusarz
Exp \$

Copyright 1999-2009 The Horde Project (<http://www.horde.org/>)

See the enclosed file COPYING for license information (LGPL). If you did not receive this file, see <http://www.fsf.org/copyleft/lgpl.html>.

- **Package** Horde_Horde_Util
- **Author** Jon Parise < jon@horde.org >
- **Author** Chuck Hagenbuch < chuck@horde.org >
- **Since** Horde 3.0

Horde_Util::\$cache

mixed = array() [line 32]

- **Static**
- **Access** protected

Horde_Util::\$files

mixed = null [line 29]

- **Static**
- **Access** protected

Horde_Util::\$securedel

mixed = null [line 30]

- **Static**
- **Access** protected

object The function Horde_Util::cloneObject(&\$obj) [line 41]

Function Parameters:

- *object* **&\$obj** The object to clone.

Returns an object's clone.

- **Access** public

void function Horde_Util::deleteAtShutdown([\$filename = false], [\$register = true], [\$secure = false]) [line 126]

Function Parameters:

- *string* **\$filename** The filename to be deleted at the end of the request.
- *boolean* **\$register** If true, then register the element for deletion, otherwise, unregister it.
- *boolean* **\$secure** If deleting file, should we securely delete the file?

Removes given elements at request shutdown.

If called with a filename will delete that file at request shutdown; if called with a directory will remove that directory and all files in that directory at request shutdown.

If called with no arguments, return all elements to be deleted (this should only be done by Horde_Util::_deleteAtShutdown).

The first time it is called, it initializes the array and registers Horde_Util::_deleteAtShutdown() as a shutdown public function - no need to do so manually.

The second parameter allows the unregistering of previously registered elements.

- **Access** public

boolean function Horde_Util::extensionExists(\$ext) [*line 197*]

Function Parameters:

- *string* **\$ext** The extension name.

Caches the result of extension_loaded() calls.

- **Access** public

string function Horde_Util::getTempFile([\$prefix = "], [\$delete = true], [\$dir = "], [\$secure = false]) [*line 77*]

Function Parameters:

- *string* **\$prefix** Prefix to make the temporary name more recognizable.
- *boolean* **\$delete** Delete the file at the end of the request?
- *string* **\$dir** Directory to create the temporary file in.
- *boolean* **\$secure** If deleting file, should we securely delete the file?

Creates a temporary filename for the lifetime of the script, and (optionally) register it to be deleted at request shutdown.

- **Access** public

void function Horde_Util::_deleteAtShutdown() [*line 163*]

Deletes registered files at request shutdown.

This public function should never be called manually; it is registered as a shutdown public function by Horde_Util::deleteAtShutdown() and called automatically at the end of the request. It will retrieve the list of folders and files to delete from Horde_Util::deleteAtShutdown()'s static array, and then iterate through, deleting folders recursively.

Contains code from gpg_public functions.php. Copyright 2002-2003 Braverock Ventures

- **Access** public

Package Horde_MIME Procedural Elements

MIME.php

- **Package** Horde_MIME

ENC7BIT = 0 *[line 21]*
ENC8BIT = 1 *[line 22]*
ENCBASE64 = 3 *[line 24]*
ENCBINARY = 2 *[line 23]*
ENCOTHER = 5 *[line 26]*
ENCQUOTEDPRINTABLE = 4 *[line 25]*
MIME_CODE = 1 *[line 39]*

Return a code for type()/encoding().

MIME_STRING = 2 *[line 44]*

Return a string for type()/encoding().

TYPEAPPLICATION = 3 *[line 14]*
TYPEAUDIO = 4 *[line 15]*
TYPEIMAGE = 5 *[line 16]*
TYPEMESSAGE = 2 *[line 13]*
TYPEMODEL = 7 *[line 33]*

Older versions of PHP's imap extension don't define TYPEMODEL.

TYPEMULTIPART = 1 *[line 12]*
TYPEOTHER = 8 *[line 18]*
TYPETEXT = 0 *[line 11]*
TYPEVIDEO = 6 *[line 17]*

Part.php

- **Package** Horde_MIME

MIME_DEFAULT_CHARSET = 'us-ascii' *[line 19]*

The default MIME character set.

MIME_DEFAULT_DESCRIPTION = 'unnamed' *[line 24]*

The default MIME description.

MIME_DEFAULT_DISPOSITION = 'inline' *[line 29]*

The default MIME disposition.

MIME_DEFAULT_ENCODING = '7bit' *[line 34]*

The default MIME encoding.

MIME_PART_EOL = "\n" *[line 6]*

The character(s) used internally for EOLs.

MIME_PART_RFC_EOL = "\r\n" *[line 12]*

The character string designated by RFCs 822/2045 to designate EOLs in MIME messages.

Package Horde_MIME Classes

Class Horde_MIME

[line 61]

The MIME:: class provides methods for dealing with MIME standards.

\$Horde: framework/MIME/MIME.php,v 1.139.4.47 2009/01/18 03:35:15 chuck Exp \$

Copyright 1999-2009 The Horde Project (<http://www.horde.org/>)

See the enclosed file COPYING for license information (LGPL). If you did not receive this file, see <http://www.fsf.org/copyleft/lgpl.html>.

- **Package** Horde_MIME
- **Author** Chuck Hagenbuch < chuck@horde.org >
- **Since** Horde 1.3

Horde_MIME::\$mime_encodings

```
array = array(  
    ENC7BIT => '7bit',  
    ENC8BIT => '8bit',  
    ENCBINARY => 'binary',  
    ENCBASE64 => 'base64',  
    ENCQUOTEDPRINTABLE => 'quoted-printable',  
    ENCOTHER => 'unknown'  
) [line 85]
```

A listing of the allowed MIME encodings.

- **Access public**

Horde_MIME::\$mime_types

```
array = array(
    TYPETEXT => 'text',
    TYPEMULTIPART => 'multipart',
    TYPEMESSAGE => 'message',
    TYPEAPPLICATION => 'application',
    TYPEAUDIO => 'audio',
    TYPEIMAGE => 'image',
    TYPEVIDEO => 'video',
    TYPEMODEL => 'model',
    TYPEOTHER => 'other'
) [line 68]
```

A listing of the allowed MIME types.

- **Access public**

Horde_MIME::\$rfc822_filter

```
string =
"()<>@,;:\\" [line 99]
```

Filter for RFC822.

- **Access public**

string function Horde_MIME::addrArray2String(\$addresses, [\$filter = "]) [line 695]

Function Parameters:

- *array* **\$addresses** The array of address objects.
- *mixed* **\$filter** A user@example.com style bare address to ignore. If any address matches \$filter, it will not be included in the final string.

Takes an array of address objects, as returned by `imap_headerinfo()`, for example, and passes each of them through `MIME::addrObject2String()`.

- **Access** public

string function Horde_MIME::addrObject2String(\$ob, [\$filter = ""]) [line 644]

Function Parameters:

- *stdClass* **\$ob** The address object to be turned into a string.
- *mixed* **\$filter** A user@example.com style bare address to ignore. Either single string or an array of strings. If the address matches \$filter, an empty string will be returned.

Takes an address object, as returned by `imap_header()` for example, and formats it as a string.

Object format for the address "John Doe <john_doe@example.com>" is:

`$object->personal` = Personal name ("John Doe")

`$object->mailbox` = The user's mailbox ("john_doe")

`$object->host` = The host the mailbox is on ("example.com")

- **Access** public

mixed function Horde_MIME::bareAddress(\$address, [\$defserver = null], [\$multiple = false]) [line 731]

Function Parameters:

- *string* **\$address** The address string.
- *string* **\$defserver** The default domain to append to mailboxes.
- *boolean* **\$multiple** Should we return multiple results?

Returns the bare address.

- **Access** public

string function Horde_MIME::decode(\$string, [\$to_charset = null]) [*line 314*]

Function Parameters:

- *string* **\$string** The text to decode.
- *string* **\$to_charset** The charset that the text should be decoded to.

Decodes an RFC 2047-encoded string.

- **Access** public

string function Horde_MIME::decodeAddrString(\$string, [\$to_charset = null]) [*line 386*]

Function Parameters:

- *string* **\$string** The text to decode.
- *string* **\$to_charset** The charset that the text should be decoded to.

Decodes an RFC 2047-encoded address string.

- **Access** public

array function Horde_MIME::decodeRFC2231(\$string, [\$to_charset = null]) [*line 461*]

Function Parameters:

- *string* **\$string** The entire string to decode, including the parameter name.
- *string* **\$to_charset** The charset the text should be decoded to.

Decodes an RFC 2231-encoded string.

- **Access** public

string function Horde_MIME::encode(\$text, [\$charset = null]) [*line 129*]

Function Parameters:

- *string* **\$text** The text to encode.
- *string* **\$charset** The character set of the text.

Encodes a string containing non-ASCII characters according to RFC 2047.

- **Access** public

string function Horde_MIME::encodeAddress(\$addresses, [\$charset = null], [\$defserver = null]) [*line 266*]

Function Parameters:

- *string* **\$addresses** The email addresses to encode.
- *string* **\$charset** The character set of the text.
- *string* **\$defserver** The default domain to append to mailboxes.

Encodes a string containing email addresses according to RFC 2047.

This differs from MIME::encode() because it keeps email addresses legal, only encoding the personal information.

- **Access** public

array function Horde_MIME::encodeRFC2231(\$name, \$string, \$charset, [\$lang = null]) [line 407]

Function Parameters:

- *string* **\$name** The parameter name.
- *string* **\$string** The string to encode.
- *string* **\$charset** The charset the text should be encoded with.
- *string* **\$lang** The language to use when encoding.

Encodes a string pursuant to RFC 2231.

- **Access** public

mixed function Horde_MIME::encoding(\$input, [\$format = null]) [line 860]

Function Parameters:

- *mixed* **\$input** Either the MIME code or encoding string.
- *integer* **\$format** If MIME_CODE, return code. If MIME_STRING, returns lowercase string. If not set, returns the opposite value.

Returns the MIME encoding for the given input.

- **Access** public

void function Horde_MIME::generateMessageID(0) [line 916]

Function Parameters:

- *string* **0** A message ID string.

Generates a Message-ID string conforming to RFC 2822 [3.6.4] and the standards outlined in 'draft-ietf-usefor-message-id-01.txt'.

- **Access** public

boolean function Horde_MIME::is8bit(\$string, [\$charset = null]) [*line 110*]

Function Parameters:

- *string* **\$string** The string to check.
- *string* **\$charset** The charset of the string. Defaults to US-ASCII. Since Horde 3.2.2.

Determines if a string contains 8-bit (non US-ASCII) characters.

- **Access** public

array function Horde_MIME::parseAddressList(\$address, [\$defserver = null], [\$validate = false]) [*line 768*]

Function Parameters:

- *string* **\$address** The address string.
- *string* **\$defserver** The default domain to append to mailboxes.
- *boolean* **\$validate** Whether to validate the address(es).

Parses a list of email addresses into its parts.

Works with and without the imap extension being available and parses distribution lists as well.

- **See** http://www.php.net/imap_rfc822_parse_adrlist
- **Since** Horde 3.2
- **Access** public

string function Horde_MIME::quotedPrintableEncode(\$text, \$eol) [*line 208*]

Function Parameters:

- *string* **\$text** The text to encode.
- *string* **\$eol** The EOL sequence to use.

Encodes a line via quoted-printable encoding.

Wraps lines at 76 characters.

- **Access** public

array function Horde_MIME::rfc822Explode(\$string, [\$delimiters = ',']) [*line 571*]

Function Parameters:

- *string* **\$string** The RFC 822 string.
- *string* **\$delimiters** A string containing valid delimiters. Defaults to ', '.

Explodes an RFC 2822 string, ignoring a delimiter if preceded by a "\" character, or if the delimiter is inside single or double quotes.

- **Access** public

string function Horde_MIME::rfc822WriteAddress(\$mailbox, [\$host = null], [\$personal = ""]) [*line 537*]

Function Parameters:

- *string* **\$mailbox** Mailbox name.
- *string* **\$host** Domain name of mailbox's host.
- *string* **\$personal** Personal name phrase.

Builds an RFC 822 compliant email address.

- **Access** public

string function Horde_MIME::trimEmailAddress(\$address) [*line 516*]

Function Parameters:

- *string* **\$address** The address to trim.

If an email address has no personal information, get rid of any angle brackets (<>) around it.

- **Access** public

mixed function Horde_MIME::type(\$input, [\$format = null]) [*line 845*]

Function Parameters:

- *mixed* **\$input** Either the MIME code or type string.
- *integer* **\$format** If MIME_CODE, return code. If MIME_STRING, returns lowercase string.

Returns the MIME type for the given input.

- **Access** public

string function Horde_MIME::wrapHeaders(\$header, \$text, [\$eol = "\r\n"]) [*line 934*]

Function Parameters:

- *string* **\$header** The header name.

- *string* **\$text** The text of the header field.
- *string* **\$eol** The EOL string to use.

Adds proper linebreaks to a header string.

RFC 2822 says headers SHOULD only be 78 characters a line, but also says that a header line MUST not be more than 998 characters.

- **Access** public

string function Horde_MIME::_encode(\$text, \$charset) [*line 173*]

Function Parameters:

- *string* **\$text** The text to encode.
- *string* **\$charset** The character set of the text.

Internal recursive public function to RFC 2047 encode a string.

- **Access** protected

mixed function Horde_MIME::_getCode(\$input, \$format, \$type) [*line 878*]

Function Parameters:

- *mixed* **\$input** Either the MIME code or encoding string.
- *string* **\$format** If MIME_CODE, returns code. If MIME_STRING, returns lowercase string. If null, returns the opposite value.
- *string* **\$type** The name of the internal array.

Retrieves MIME encoding/type data from the internal arrays.

- **Access** protected

string function Horde_MIME::_rfc822Encode(\$str, [\$type = 'address']) [*line 800*]

Function Parameters:

- *string* **\$str** The string to be quoted and escaped.
- *string* **\$type** Either 'address' or 'personal'.

Quotes and escapes the given string if necessary using rules contained in RFC 2822 [3.2.5].

- **Access** protected

Class Horde_MIME_Message

[*line 18*]

The MIME_Message:: class provides methods for creating and manipulating MIME email messages.

\$Horde: framework/MIME/MIME/Message.php,v 1.76.10.19 2009/01/06 15:23:20 jan Exp
\$

Copyright 1999-2009 The Horde Project (<http://www.horde.org/>)

See the enclosed file COPYING for license information (LGPL). If you did not receive this file, see <http://www.fsf.org/copyleft/lgpl.html>.

- **Package** Horde_MIME
- **Author** Chuck Hagenbuch < chuck@horde.org >
- **Author** Michael Slusarz < slusarz@horde.org >

Horde_MIME_Message::\$_build

boolean = false [line 25]

Has the message been parsed via buildMessage()?

- **Access** protected

Horde_MIME_Message::\$_defaultServer

string = null [line 32]

The server to default unqualified addresses to.

- **Access** protected

Constructor *void* function Horde_MIME_Message::__construct([\$defaultServer = null]) [line 40]

Function Parameters:

- *string* **\$defaultServer** The server to default unqualified addresses to.

Constructor - creates a new MIME email message.

- **Access** public

void function Horde_MIME_Message::buildMessage() [line 141]

Build message from current contents.

- **Access** public

MIME_Message function Horde_MIME_Message::convertMIMEPart(&\$mime_part, [\$server = null]) [*line 60*]

Function Parameters:

- *MIME_Part* **&\$mime_part** The MIME_Part object.
- *string* **\$server** The server to default unqualified addresses to.

Create a MIME_Message object from a MIME_Part object.

This public function can be called statically via: `MIME_Message::convertMIMEPart();`

- **Access** public

array function Horde_MIME_Message::encode(\$headers, \$charset) [*line 81*]

Function Parameters:

- *array* **\$headers** The headers to encode.
- *string* **\$charset** The character set to use.

Take a set of headers and make sure they are encoded properly.

- **Access** public

MIME_Message function Horde_MIME_Message::getBasePart() [*line 187*]

Return the base part of the message. This public function does NOT return a reference to make sure that the whole MIME_Message object isn't accidentally modified.

- **Access** public

MIME_Part function Horde_MIME_Message::getPart(\$id) [*line 201*]

Function Parameters:

- *string* **\$id** The MIME_Part ID string.

Retrieve a specific MIME part.

- **Access** public

array function Horde_MIME_Message::getParts() [*line 170*]

Get a list of all MIME subparts.

- **Access** public

array function Horde_MIME_Message::header([\$headers = array()]) [*line 110*]

Function Parameters:

- *array* **\$headers** The headers to add the MIME headers to.

Add the proper set of MIME headers for this message to an array.

- **Access** public

string function Horde_MIME_Message::toString([\$headers = false]) *[line 128]*

Function Parameters:

- **\$headers**

Return the entire message contents, including headers, as a string.

- **Access** public

Class Horde_MIME_Part

[line 51]

The MIME_Part:: class provides a wrapper around MIME parts and methods for dealing with them.

\$Horde: framework/MIME/MIME/Part.php,v 1.177.4.27 2009/06/17 19:49:13 slusarz Exp
\$

Copyright 1999-2009 The Horde Project (<http://www.horde.org/>)

See the enclosed file COPYING for license information (LGPL). If you did not receive this file, see <http://www.fsf.org/copyleft/lgpl.html>.

- **Package** Horde_MIME
- **Author** Chuck Hagenbuch < chuck@horde.org >
- **Author** Michael Slusarz < slusarz@horde.org >

Horde_MIME_Part::\$_boundary

string = null *[line 183]*

Unique MIME_Part boundary string.

- **Access** protected

Horde_MIME_Part::\$_bytes

integer = 0 [line 190]

Default value for this Part's size.

- **Access** protected

Horde_MIME_Part::\$_cids

array = array() [line 144]

The list of CIDs for this part.

- **Access** protected

Horde_MIME_Part::\$_contentid

string = null [line 197]

The content-ID for this part.

- **Access** protected

Horde_MIME_Part::\$_contents

string = " [line 74]

The body of the part.

- **Access** protected

Horde_MIME_Part::\$_contentTypeParameters

array = array() [line 123]

The content type parameters of this part.

- **Access** protected

Horde_MIME_Part::\$_currentEncoding

string = null [line 88]

The current transfer encoding of the contents of this part.

- **Access** protected

Horde_MIME_Part::\$_description

string = " [line 102]

The description of this part.

- **Access** protected

Horde_MIME_Part::\$_disposition

string = MIME_DEFAULT_DISPOSITION [*line 109*]

The disposition of this part (inline or attachment).

- **Access** protected

Horde_MIME_Part::\$_dispositionParameters

array = array() [*line 116*]

The disposition parameters of this part.

- **Access** protected

Horde_MIME_Part::\$_encode7bit

boolean = true [*line 95*]

Should the message be encoded via 7-bit?

- **Access** protected

Horde_MIME_Part::\$_eol

string = MIME_PART_EOL [*line 162*]

The sequence to use as EOL for this part.

The default is currently to output the EOL sequence internally as just "\n"

instead of the canonical "\r\n" required in RFC 822 & 2045. To be RFC compliant, the full <CR><LF> EOL combination should be used when sending a message.

- **Access** protected

Horde_MIME_Part::\$_flags

array = array() [line 169]

Internal class flags.

- **Access** protected

Horde_MIME_Part::\$_idmap

array = array() [line 176]

Part -> ID mapping cache.

- **Access** protected

Horde_MIME_Part::\$_information

array = array() [line 137]

Information/Statistics on the subpart.

- **Access** protected

Horde_MIME_Part::\$mimeid

string = null [line 151]

The MIME ID of this part.

- **Access** protected

Horde_MIME_Part::\$parts

array = array() [line 130]

The subparts of this part.

- **Access** protected

Horde_MIME_Part::\$subtype

string = 'octet-stream' [line 67]

The subtype (ex.: plain) of this part.

Per RFC 2045, the default is 'octet-stream'.

- **Access** protected

Horde_MIME_Part::\$transferEncoding

string = MIME_DEFAULT_ENCODING [line 81]

The desired transfer encoding of this part.

- **Access** protected

Horde_MIME_Part::\$_type

string = 'application' [*line 59*]

The type (ex.: text) of this part.

Per RFC 2045, the default is 'application'.

- **Access** protected

Constructor *void* function Horde_MIME_Part::__construct([\$mimetype = null], [\$contents = null], [\$charset = MIME_DEFAULT_CHARSET], [\$disposition = null], [\$encoding = null]) [*line 208*]

Function Parameters:

- *string* **\$mimetype** The content type of the part.
- *string* **\$contents** The body of the part.
- *string* **\$charset** The character set of the part.
- *string* **\$disposition** The content disposition of the part.
- *string* **\$encoding** The content encoding of the contents.

MIME_Part constructor.

- **Access** public

void function Horde_MIME_Part::addCID([\$cids = array()]) [*line 1336*]

Function Parameters:

- *array* **\$cids** A list of MIME IDs of the part.
for the part
- | | |
|---------------|-------------|
| Key - MIME ID | Value - CID |
|---------------|-------------|

Add to the list of CIDs for this part.

- **Access** public

void function Horde_MIME_Part::addPart(\$mime_part, [\$index = null]) [*line 569*]

Function Parameters:

- *MIME_Part* **\$mime_part** Add a MIME_Part subpart to the current MIME_Part.
- *string* **\$index** The index of the added MIME_Part.

Add a MIME subpart.

- **Access** public

void function Horde_MIME_Part::alterPart(\$id, \$mime_part) [*line 669*]

Function Parameters:

- *string* **\$id** The MIME Part ID to alter.
- *MIME_Part* **\$mime_part** The MIME Part to store.

Alter a current MIME subpart.

- **Access** public

void function Horde_MIME_Part::appendContents(\$contents, [\$encoding = null]) [*line 314*]

Function Parameters:

- *string* **\$contents** The contents to append to the current part body.
- *string* **\$encoding** The current encoding of the contents. If not specified, will try to auto determine the encoding.

Add to the body contents of this part.

- **Access public**

void function Horde_MIME_Part::clearContents() [*line 322*]

Clears the body contents of this part.

- **Access public**

void function Horde_MIME_Part::clearContentTypeParameter(\$label, \$data) [*line 810*]

Function Parameters:

- *string* **\$label** The disposition parameter label.
- *string* **\$data** The disposition parameter data.

Clears a content type parameter from this part.

- **Access public**

array function Horde_MIME_Part::contentTypeMap() [*line 1416*]

Returns a mapping of all MIME IDs to their content-types.

- **Access** public

array function Horde_MIME_Part::getAllContentTypeParameters() [*line 834*]

Get all parameters from the Content-Type header.

- **Access** public

array function Horde_MIME_Part::getAllDispositionParameters() [*line 788*]

Get all parameters from the Content-Disposition header.

- **Access** public

integer function Horde_MIME_Part::getBytes() [*line 1288*]

Determine the size of a MIME_Part and its child members.

- **Access** public

string function Horde_MIME_Part::getCanonicalContents() [*line 345*]

Returns the contents in strict RFC 822 & 2045 output - namely, all newlines end with the canonical <CR><LF> sequence.

- **Access** public

string function Horde_MIME_Part::getCharset() [*line 500*]

Get the character set to use for of this part. Returns a charset for all types (not just 'text/*') since we use this charset to determine how to encode text in MIME headers.

- **Access public**

array function Horde_MIME_Part::getCIDList() [*line 1346*]

Returns the list of CIDs for this part.

- **Access public**

string function Horde_MIME_Part::getContentID() [*line 1371*]

Returns the Content-ID for this part.

- **Access public**

string function Horde_MIME_Part::getContents() [*line 334*]

Return the body of the part.

- **Access public**

string function Horde_MIME_Part::getContentTypeParameter(\$label) [*line 823*]

Function Parameters:

- *string* **\$label** The content type parameter label.

Get a content type parameter from this part.

- **Access** public

string function Horde_MIME_Part::getCurrentEncoding() [*line 1085*]

Retrieves the current encoding of the contents in the object.

- **Access** public

string function Horde_MIME_Part::getDefaultType() [*line 444*]

If the subtype of a MIME part is unrecognized by an application, the default type should be used instead (See RFC 2046). This method returns the default subtype for a particular primary MIME Type.

- **Access** public

string function Horde_MIME_Part::getDescription([\$decode = false], [\$default = false]) [*line 526*]

Function Parameters:

- *boolean* **\$decode** MIME decode description?
- *boolean* **\$default** If the name parameter doesn't exist, should we use the default name from the description parameter?

Get the description of this part.

- **Access** public

string function Horde_MIME_Part::getDisposition() [*line 250*]

Get the content-disposition of this part.

- **Access** public

string function Horde_MIME_Part::getDispositionParameter(\$label) [*line 777*]

Function Parameters:

- *string* **\$label** The disposition parameter label.

Get a disposition parameter from this part.

- **Access** public

string function Horde_MIME_Part::getEOL() [*line 854*]

Get the string to use for EOLs.

- **Access** public

mixed function Horde_MIME_Part::getInformation(\$label) [*line 753*]

Function Parameters:

- *string* **\$label** The information label.

Retrieve information about the MIME_Part.

- **Access** public

string function Horde_MIME_Part::getMIMEId() [*line 1391*]

Returns the MIME ID of this part.

- **Access** public

string function Horde_MIME_Part::getName([\$decode = false], [\$default = false]) [*line 277*]

Function Parameters:

- *boolean* **\$decode** MIME decode description?
- *boolean* **\$default** If the name parameter doesn't exist, should we use the default name from the description parameter?

Get the name of this part.

- **Access** public

MIME_Part function Horde_MIME_Part::getPart(\$id) [*line 615*]

Function Parameters:

- *string* **\$id** The MIME_Part ID string.

Retrieve a specific MIME part.

- **Access** public

array function Horde_MIME_Part::getParts() [*line 602*]

Get a list of all MIME subparts.

- **Access** public

string function Horde_MIME_Part::getPrimaryType() [*line 467*]

Get the primary type of this part.

- **Access** public

string function Horde_MIME_Part::getRelativeMIMEId(\$id) [*line 1405*]

Function Parameters:

- *string* **\$id** The relative part ID.

Returns the relative MIME ID of this part.

e.g., if the base part has MIME ID of 2, and you want the first subpart of the base part, the relative MIME ID is 2.1.

- **Access** public

string function Horde_MIME_Part::getSubType() [*line 477*]

Get the subtype of this part.

- **Access** public

string function Horde_MIME_Part::getTransferEncoding() [*line 1026*]

Get the transfer encoding for the part based on the user requested transfer encoding and the current contents of the part.

- **Access** public

string function Horde_MIME_Part::getType([\$charset = false]) [*line 424*]

Function Parameters:

- *boolean* **\$charset** Append character set information to the end of the content type if this is a text/* part.

Get the full MIME Content-Type of this part.

- **Access** public

string function Horde_MIME_Part::getUniqueID() [*line 1278*]

Return the unique MIME_Part boundary string generated for this object.

This may not be the boundary string used when building the message since a user defined 'boundary' Content-Type parameter will override this value.

- **Access** public

array function Horde_MIME_Part::header([\$headers = array()]) [*line 866*]

Function Parameters:

- *array* **\$headers** An array of any other headers for the part.

Add the appropriate MIME headers for this part to an existing array.

- **Access** public

void function Horde_MIME_Part::removePart(\$id) [*line 655*]

Function Parameters:

- *string* **\$id** The MIME Part to delete.

Remove a MIME_Part subpart.

- **Access** public

string function Horde_MIME_Part::replaceEOL(\$text, [\$eol = null]) [*line 1262*]

Function Parameters:

- *string* **\$text** The text to replace.
- *string* **\$eol** The EOL sequence to use. If not present, uses the part's current EOL setting.

Replace newlines in this part's contents with those specified by either the given newline sequence or the part's current EOL setting.

- **Access** public

void function Horde_MIME_Part::setBytes(\$bytes) [*line 1324*]

Function Parameters:

- *integer* **\$bytes** The size of this part in bytes.

Explicitly set the size (in bytes) of this part. This value will only be returned (via `getBytes()`) if there are no contents currently set.

This public function is useful for setting the size of the part when the contents of the part are not fully loaded (i.e. creating a `MIME_Part` object from IMAP header information without loading the data of the part).

- **Access** public

void function Horde_MIME_Part::setCharset(\$charset) [*line 487*]

Function Parameters:

- *string* **\$charset** The character set of this part.

Set the character set of this part.

- **Access** public

void function Horde_MIME_Part::setContentID([\$cid = null]) [*line 1358*]

Function Parameters:

- *string* **\$cid** Use this CID (if not already set). Else, generate a random CID.

Sets the Content-ID header for this part.

- **Access** public

void function Horde_MIME_Part::setContents(\$contents, [\$encoding = null]) *[line 298]*

Function Parameters:

- *string* **\$contents** The part body.
- *string* **\$encoding** The current encoding of the contents.

Set the body contents of this part.

- **Access** public

void function Horde_MIME_Part::setContentParameter(\$label, \$data) *[line 799]*

Function Parameters:

- *string* **\$label** The disposition parameter label.
- *string* **\$data** The disposition parameter data.

Add a content type parameter to this part.

- **Access** public

void function Horde_MIME_Part::setDescription(\$description) [*line 511*]

Function Parameters:

- *string* **\$description** The description of this part.

Set the description of this part.

- **Access** public

void function Horde_MIME_Part::setDisposition(\$disposition) [*line 236*]

Function Parameters:

- *string* **\$disposition** The content-disposition to set (inline or attachment).

Set the content-disposition of this part.

- **Access** public

void function Horde_MIME_Part::setDispositionParameter(\$label, \$data) [*line 764*]

Function Parameters:

- *string* **\$label** The disposition parameter label.
- *string* **\$data** The disposition parameter data.

Add a disposition parameter to this part.

- **Access** public

void function Horde_MIME_Part::setEOL(\$eol) [line 844]

Function Parameters:

- *string* **\$eol** The string to use for EOLs.

Sets a new string to use for EOLs.

- **Access** public

void function Horde_MIME_Part::setInformation(\$label, \$data) [line 740]

Function Parameters:

- *string* **\$label** The information label.
- *mixed* **\$data** The information to store.

Add information about the MIME_Part.

- **Access** public

void function Horde_MIME_Part::setMIMEId(\$mimeid) [line 1381]

Function Parameters:

- *string* **\$mimeid** The MIME ID.

Alter the MIME ID of this part.

- **Access** public

void function Horde_MIME_Part::setName(\$name) [*line 262*]

Function Parameters:

- *string* **\$name** The name to set.

Set the name of this part.

TODO: MIME encode here instead of in header() - add a charset parameter.

- **Access** public

void function Horde_MIME_Part::setTransferEncoding(\$encoding) [*line 549*]

Function Parameters:

- *string* **\$encoding** The transfer encoding to use.

Set the transfer encoding to use for this part.

- **Access** public

void function Horde_MIME_Part::setType(\$mimetype) [*line 385*]

Function Parameters:

- *string* **\$mimetype** The mimetype to set (ex.: text/plain).

Set the mimetype of this part.

- **Access** public

boolean function Horde_MIME_Part::splitContents() [*line 1202*]

Split the contents of the current Part into its respective subparts, if it is multipart MIME encoding. Unlike the imap_*() public functions, this will preserve all MIME header information.

The boundary content-type parameter must be set for this public function to work correctly.

- **Access** public

void function Horde_MIME_Part::strict7bit(\$use7bit) [*line 1015*]

Function Parameters:

- *boolean* **\$use7bit** Use 7-bit encoding?

Should we make sure the message is encoded via 7-bit (e.g. to adhere to mail delivery standards such as RFC 2821)?

- **Access** public

string function Horde_MIME_Part::toCanonicalString([\$headers = true]) [*line 1003*]

Function Parameters:

- *boolean* **\$headers** Include the MIME headers?

Returns the encoded part in strict RFC 822 & 2045 output - namely, all newlines end with the canonical <CR><LF> sequence.

- **Access** public

string function Horde_MIME_Part::toString([\$headers = true]) [*line 945*]

Function Parameters:

- *boolean* **\$headers** Include the MIME headers?

Return the entire part in MIME format. Includes headers on request.

- **Access** public

string function Horde_MIME_Part::transferDecode() [*line 1140*]

Decodes the contents of the part to either a 7bit or 8bit encoding.

- **Access** public

void function Horde_MIME_Part::transferDecodeContents() [*line 365*]

Transfer decode the contents and set them as the new contents.

- **Access** public

string function Horde_MIME_Part::transferEncode() [*line 1095*]

Encodes the contents with the part's transfer encoding.

- **Access** public

void function Horde_MIME_Part::transferEncodeContents() [*line 354*]

Transfer encode the contents (to the transfer encoding identified via getTransferEncoding()) and set as the part's new contents.

- **Access** public

string function Horde_MIME_Part::_generateBoundary() [*line 1432*]

Generate the unique boundary string (if not already done).

- **Access** protected

void function Horde_MIME_Part::_generateIdMap(&\$parts) [*line 723*]

Function Parameters:

- *array* **&\$parts** An array of MIME_Parts to map.

Generates a mapping of MIME_Parts with their MIME IDs.

- **Access** protected

void function Horde_MIME_Part::_partFind(\$id, &\$parts, [\$retarray = false]) [*line 688*]

Function Parameters:

- *string* **\$id** The MIME_Part ID string.
- *array* **&\$parts** A list of MIME_Part objects.
- *boolean* **\$retarray** Return a pointer to the array that stores (would store) the part rather than the part itself?

Function used to find a specific MIME Part by ID.

- **Access** public

Class Horde_MIME_Structure

[*line 23*]

\$Horde: framework/MIME/MIME/Structure.php,v 1.87.10.31 2009/01/06 15:23:20 jan Exp \$

The MIME_Structure:: class provides methods for dealing with MIME mail.

The default character set to use for messages should be defined in the variable \$GLOBALS['mime_structure']['default_charset'] (defaults to US-ASCII per RFC 2045).

TODO: Convert to OO

Copyright 1999-2009 The Horde Project (<http://www.horde.org/>)

See the enclosed file COPYING for license information (LGPL). If you did not receive this file, see <http://www.fsf.org/copyleft/lgpl.html>.

- **Package** Horde_MIME
- **Author** Chuck Hagenbuch < chuck@horde.org >
- **Author** Michael Slusarz < slusarz@horde.org >

void function Horde_MIME_Structure::addMultipartInfo(&\$parts, [\$info = array()]) [line 247]

Function Parameters:

- *array* **&\$parts** The list of parts contained within the multipart object.
- *array* **\$info** Information about the multipart structure.

Set the special information for certain MIME types.

- **Since** Horde 3.2
- **Static**
- **Access** public

&MIME_Message function Horde_MIME_Structure::parse(\$body) [line 33]

Function Parameters:

- *stdClass* **\$body** The result of `imap_fetchstructure()`.

Given the results of `imap_fetchstructure()`, parse the structure of the message, figuring out correct bodypart numbers, etc.

- **Static**
- **Access** public

array function Horde_MIME_Structure::parseMIMEHeaders(\$headers, [\$decode = true], [\$lowercase = false]) [line 415]

Function Parameters:

- *string* **\$headers** A text string containing the headers (e.g. `imap_fetchheader()`) output from
- *boolean* **\$decode** Should the headers be decoded?
- *boolean* **\$lowercase** Should the keys be in lowercase?

Builds an array consisting of MIME header/value pairs.

- **Static**
- **Access** public

MIME_Message function Horde_MIME_Structure::parseTextMIMEMessage(\$text) [*line 297*]

Function Parameters:

- *string* **\$text** The text of the MIME message.

Attempts to build a MIME_Message object from a text message.

- **Static**
- **Access** public

void function Horde_MIME_Structure::_convertMimeDecodeData(&\$ob) [*line 326*]

Function Parameters:

- *stdClass* **&\$ob** The output from mimeDecode::decode().

Convert the output from mimeDecode::decode() into a structure that matches imap_fetchstructure() output.

- **Static**
- **Access** protected

array function Horde_MIME_Structure::_getParameters(\$part, \$type) [*line 194*]

Function Parameters:

- *stdClass* **\$part** The original part info.
- *integer* **\$type** The parameter type to retrieve.
 1. = content
 2. = disposition

Get all parameters for a given portion of a message.

- **Static**
- **Access** protected

MIME_Part function Horde_MIME_Structure::_parse(\$body, [\$ref = 0]) [*line 56*]

Function Parameters:

- *stdClass* **\$body** The result of `imap_fetchstructure()`.
- *string* **\$ref** The current bodypart.

Given the results of `imap_fetchstructure()`, parse the structure of the message, figuring out correct bodypart numbers, etc.

- **Static**
- **Access** protected

void function Horde_MIME_Structure::_setInfo(\$part, &\$ob, \$ref) [*line 115*]

Function Parameters:

- *stdClass* **\$part** The original part info.
- *MIME_Part* **&\$ob** A *MIME_Part* object.
- *string* **\$ref** The ID of this part.

Given a mime part from `imap_fetchstructure()`, munge it into a useful form and make sure that any parameters which are missing are given default values.

To specify the default character set, define the global variable `$GLOBALS['mime_structre']['default_charset']`.

- **Static**
- **Access** protected

Package AS2Secure Classes

Class AS2Abstract

[line 32]

AS2Secure - PHP Lib for AS2 message encoding / decoding

- **Package** AS2Secure
- **Author** Sebastien MALOT < contact@as2secure.com >
- **Version** 0.8.4
- **Copyright** Copyright (c) 2010, Sebastien MALOT

Last release at : <http://www.as2secure.com>

This file is part of AS2Secure Project.

AS2Secure is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

AS2Secure is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with AS2Secure.

- **Abstract Element**
- **License** [GNU General Public License](#)

AS2Abstract::\$adapter

mixed = null [line 34]

- **Access** protected

AS2Abstract::\$filename

mixed = null [*line 36*]

- **Access** protected

AS2Abstract::\$files

mixed = array() [*line 39*]

- **Access** protected

AS2Abstract::\$headers

mixed = null [*line 40*]

- **Access** protected

AS2Abstract::\$is_crypted

mixed = false [*line 44*]

- **Access** protected

AS2Abstract::\$is_signed

mixed = false [*line 43*]

- **Access** protected

AS2Abstract::\$message_id

mixed = " [line 41]

- **Access** protected

AS2Abstract::\$mimetype

mixed = null [line 37]

- **Access** protected

AS2Abstract::\$partner_from

mixed = null [line 46]

- **Access** protected

AS2Abstract::\$partner_to

mixed = null [line 47]

- **Access** protected

AS2Abstract::\$path

mixed = null [line 38]

- **Access** protected

Constructor *void* function AS2Abstract::__construct(\$data, [\$params = array()]) [*line 49*]

Function Parameters:

- **\$data**
- **\$params**

- **Access** public

void function AS2Abstract::addFile(\$file) [*line 109*]

Function Parameters:

- **\$file**

- **Access** public

void function AS2Abstract::decode() [*line 163*]

- **Access** public

void function AS2Abstract::encode() [*line 159*]

- **Access** public

void function AS2Abstract::generateMessageID(\$partner) [line 175]

Function Parameters:

- **\$partner**
 - **Static**
 - **Access** protected

void function AS2Abstract::getAuthentication() [line 137]

- **Access** public

void function AS2Abstract::getContent() [line 121]

- **Access** public

void function AS2Abstract::getFileName() [line 117]

- **Access** public

void function AS2Abstract::getFiles() [line 113]

- **Access** public

void function AS2Abstract::getHeader(\$token) [line 133]

Function Parameters:

- **\$token**

- **Access** public

void function AS2Abstract::getHeaders() [*line 129*]

- **Access** public

void function AS2Abstract::getMessageId() [*line 147*]

- **Access** public

void function AS2Abstract::getPartnerFrom() [*line 92*]

- **Access** public

void function AS2Abstract::getPartnerTo() [*line 100*]

- **Access** public

void function AS2Abstract::getPath() [*line 105*]

- **Access** public

void function AS2Abstract::getUrl() [*line 167*]

- **Access** public

void function AS2Abstract::isCrypted() [line 151]

- **Access** public

void function AS2Abstract::isSigned() [line 155]

- **Access** public

void function AS2Abstract::setHeaders(\$headers) [line 125]

Function Parameters:

- **\$headers**

- **Access** public

void function AS2Abstract::setMessageId(\$id) [line 143]

Function Parameters:

- **\$id**

- **Access** public

void function AS2Abstract::setPartnerFrom(\$partner_from) [*line 88*]

Function Parameters:

- **\$partner_from**

- **Access** public

void function AS2Abstract::setPartnerTo(\$partner_to) [*line 96*]

Function Parameters:

- **\$partner_to**

- **Access** public

Class AS2Adapter

[*line 32*]

AS2Secure - PHP Lib for AS2 message encoding / decoding

- **Package** AS2Secure
- **Author** Sebastien MALOT < contact@as2secure.com >
- **Version** 0.8.4
- **Copyright** Copyright (c) 2010, Sebastien MALOT

Last release at : <http://www.as2secure.com>

This file is part of AS2Secure Project.

AS2Secure is free software: you can redistribute it and/or modify it under the terms of the GNU

General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

AS2Secure is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with AS2Secure.

- **License** [GNU General Public License](#)

AS2Adapter::\$javapath

mixed = 'java' [*line 39*]

- **Static**
- **Access** public

AS2Adapter::\$partner_from

mixed = null [*line 41*]

- **Access** protected

AS2Adapter::\$partner_to

mixed = null [*line 42*]

- **Access** protected

AS2Adapter::\$ssl_adapter

mixed = 'AS2Secure.jar' [*line 37*]

Allow to specify full path to main applications
for overriding PATH usage

- **Static**
- **Access** public

AS2Adapter::\$ssl_openssl

mixed = 'openssl' [*line 38*]

- **Static**
- **Access** public

AS2Adapter::\$tmp_files

mixed = null [*line 47*]

Array to store temporary files created and scheduled to unlink

- **Static**
- **Access** protected

Constructor *void* function AS2Adapter::__construct(\$partner_from, \$partner_to) [*line 49*]

Function Parameters:

- **\$partner_from**
- **\$partner_to**

- **Access** public

void function AS2Adapter::addTempFileForDelete(\$file) [*line 503*]

Function Parameters:

- **\$file**

Schedule file for deletion

- **Static**
- **Access** public

string function AS2Adapter::calculateMicChecksum(\$input, [\$algo = 'sha1']) [*line 384*]

Function Parameters:

- *string* **\$input** The file to use
- *string* **\$algo** The algo to use

Calculate the message integrity check (MIC) using SHA1 or MD5 algo

- **Static**
- **Access** public

string function AS2Adapter::compose(\$files, \$input) [*line 72*]

Function Parameters:

- *array* **\$input** The list of extracted files (path / mimetype / filename)
- **\$files**

Generate a mime multipart file from files

- **Access** public

string function AS2Adapter::compress(\$input) [*line 151*]

Function Parameters:

- *string* **\$input** The file to compress

Compress a file which contains mime headers

- **Access** public

string function AS2Adapter::decompress(\$input) [*line 177*]

Function Parameters:

- *string* **\$input** The file to compress

Decompress a file which contains mime headers

- **Access** public

string function AS2Adapter::decrypt(\$input) [*line 337*]

Function Parameters:

- *string* **\$input** The file to decrypt

Decrypt a message

- **Access** public

string function AS2Adapter::detectMimeType(\$file) [*line 593*]

Function Parameters:

- *string* **\$file** The file to analyse

Determine the mimetype of a file (also called 'Content-Type')

- **Static**
- **Access** public

string function AS2Adapter::detectOS() [*line 687*]

Determine the Server OS

- **Static**
- **Access** public

string function AS2Adapter::encrypt(\$input, [\$cypher = 'des3']) [*line 278*]

Function Parameters:

- *string* **\$input** The file to encrypt
- *string* **\$cypher** The Cypher to use for encryption

Encrypt a file

- **Access** public

string function AS2Adapter::exec(\$command, [\$return_output = false]) [*line 530*]

Function Parameters:

- *string* **\$command** The command line to execute
- *boolean* **\$return_output** True to return all data from standard output
False to return only the error code

Execute a command line and throw Exception if an error appends

- **Static**
- **Access** public

array function AS2Adapter::extract(\$input) [*line 108*]

Function Parameters:

- *string* **\$input** The file to extract

Extract files from mime multipart file

- **Access** public

string function AS2Adapter::getCAFromPKCS12(\$input, [\$password = ""]) [*line 447*]

Function Parameters:

- *string* **\$input** The PKCS12 Certificate
- *string* **\$password** The PKCS12 Certificate's password

Extract CA from a PKCS12 Certificate

- **Static**
- **Access** public

string function AS2Adapter::getDataFromPKCS12(\$input, \$token, [\$password = "]) [*line 460*]

Function Parameters:

- *string* **\$input** The PKCS12 Certificate
- *string* **\$token** The Part to extract
- *string* **\$password** The PKCS12 Certificate's password

Extract Part from a PKCS12 Certificate

- **Static**
- **Access** protected

string function AS2Adapter::getMicChecksum(\$input) [*line 398*]

Function Parameters:

- *string* **\$input** The file containing the signed message

Extract the message integrity check (MIC) from the digital signature

- **Static**
- **Access** public

string function AS2Adapter::getPrivateFromPKCS12(\$input, [\$password = "]) [*line 423*]

Function Parameters:

- *string* **\$input** The PKCS12 Certificate
- *string* **\$password** The PKCS12 Certificate's password

Extract Private Certificate from a PKCS12 Certificate

- **Static**
- **Access** public

string function AS2Adapter::getPublicFromPKCS12(\$input, [\$password = "]) [*line 435*]

Function Parameters:

- *string* **\$input** The PKCS12 Certificate
- *string* **\$password** The PKCS12 Certificate's password

Extract Public certificate from a PKCS12 Certificate

- **Static**
- **Access** public

string function AS2Adapter::getTempFilename() [*line 487*]

Create a temporary file into temporary directory and add it to the garbage collector at shutdown

- **Static**
- **Access** public

string function AS2Adapter::hex2bin(\$str) [*line 576*]

Function Parameters:

- *string* **\$str** The string in hexadecimal format.

Convert a string from hexadecimal format to binary format

- **Static**
- **Access** public

string function AS2Adapter::sign(\$input, [\$use_zlib = false], [\$encoding = 'base64']) [*line 205*]

Function Parameters:

- *string* **\$input** The file to sign
- *boolean* **\$use_zlib** Use Zlib to compress main container
- *string* **\$encoding** Encoding to use for main container (base64 | binary)

Sign a file which contains mime headers

- **Access** public

string function AS2Adapter::verify(\$input) [*line 243*]

Function Parameters:

- *string* **\$input** The file to check

Verify a signed file

- **Access** public

void function AS2Adapter::_deleteTempFiles() [*line 516*]

Garbage collector to delete temp files created with 'self::getTempFilename'
with shutdown function

- **Static**
- **Access** public

Class AS2Client

[*line 32*]

AS2Secure - PHP Lib for AS2 message encoding / decoding

- **Package** AS2Secure
- **Author** Sebastien MALOT < contact@as2secure.com >
- **Version** 0.8.4
- **Copyright** Copyright (c) 2010, Sebastien MALOT

Last release at : <http://www.as2secure.com>

This file is part of AS2Secure Project.

AS2Secure is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

AS2Secure is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with AS2Secure.

- **License** [GNU General Public License](#)

AS2Client::\$response_content

mixed = " *[line 34]*

- **Access** protected

AS2Client::\$response_headers

mixed = array() *[line 33]*

- **Access** protected

AS2Client::\$response_indice

mixed = 0 *[line 35]*

- **Access** protected

Constructor *void* function AS2Client::__construct() *[line 37]*

- **Access** public

array function AS2Client::getLastResponse() *[line 118]*

Return the last request : headers/content

- **Access** public

int function AS2Client::handleResponseHeader(\$curl, \$header) [*line 132*]

Function Parameters:

- *object* **\$curl** The curl instance
- *string* **\$header** The header received

Allow to retrieve HTTP headers even if there is HTTP redirections

- **Access** protected

array function AS2Client::sendRequest(\$request) [*line 48*]

Function Parameters:

- [AS2Abstract](#) **\$request** The request to send (instanceof : AS2Message | AS2MDN)

Send request to the partner (manage headers, security, ...)

- **Access** public

Class AS2Connector

[*line 32*]

AS2Secure - PHP Lib for AS2 message encoding / decoding

- **Package** AS2Secure

- **Author** Sebastien MALOT < contact@as2secure.com >
- **Version** 0.8.4
- **Copyright** Copyright (c) 2010, Sebastien MALOT

Last release at : <http://www.as2secure.com>

This file is part of AS2Secure Project.

AS2Secure is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

AS2Secure is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with AS2Secure.

- **License** [GNU General Public License](#)

AS2Connector::STATUS_ERROR

= 'ERROR' [*line 34*]

AS2Connector::STATUS_OK

= 'OK' [*line 33*]

void function AS2Connector::onReceivedMDN(\$from, \$to, \$status, \$mdn) [line 36]

Function Parameters:

- **\$from**
- **\$to**
- **\$status**
- **\$mdn**

- **Static**
- **Access** public

void function AS2Connector::onReceivedMessage(\$from, \$to, \$status, \$message) [line 41]

Function Parameters:

- **\$from**
- **\$to**

- **\$status**
- **\$message**

- **Static**
- **Access** public

void function AS2Connector::onSentMDN(\$from, \$to, \$status, \$mdn) [line 46]

Function Parameters:

- **\$from**
- **\$to**
- **\$status**
- **\$mdn**

- **Static**
- **Access** public

void function AS2Connector::onSentMessage(\$from, \$to, \$status, \$message) [line 51]

Function Parameters:

- **\$from**
- **\$to**
- **\$status**
- **\$message**

- **Static**
- **Access** public

Class AS2Exception

[line 32]

AS2Secure - PHP Lib for AS2 message encoding / decoding

- **Package** AS2Secure
- **Author** Sebastien MALOT < contact@as2secure.com>
- **Version** 0.8.4
- **Copyright** Copyright (c) 2010, Sebastien MALOT

Last release at : <http://www.as2secure.com>

This file is part of AS2Secure Project.

AS2Secure is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

AS2Secure is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with AS2Secure.

- **License** [GNU General Public License](#)

AS2Exception::DEFAULT_ERROR

= 6 *[line 61]*

AS2Exception::STATUS_ERROR

= 'error' *[line 38]*

Refers to RFC 4130

<http://rfclibrary.hosting.com/rfc/rfc4130/rfc4130-34.asp>

AS2Exception::STATUS_FAILURE

= 'failure' *[line 39]*

AS2Exception::STATUS_WARNING

= 'warning' *[line 40]*

AS2Exception::\$level_error

```
mixed = array(
    1 => 'authentication-failed',    // the receiver could not authenticate the sender
    2 => 'decompression-failed',    //
    3 => 'decryption-failed',       // the receiver could not decrypt the message contents
    4 => 'insufficient-message-security', //
    5 => 'integrity-check-failed',  // the receiver could not verify content integrity
    6 => 'unexpected-processing-error', // a catch-all for any additional processing errors
) [line 42]
```

- **Static**
- **Access** protected

AS2Exception::\$level_failure

```
mixed = array(
    101 => 'unsupported format',    // sha1, md5
    102 => 'unsupported MIC-algorithms', //
) [line 51]
```

- **Static**
- **Access** protected

AS2Exception::\$level_warning

```
mixed = array(
    201 => 'duplicate-document',    // an identical message already exists at the destination server
    202 => 'sender-equals-receiver', // the AS2-To name is identical to the AS2-From name
) [line 56]
```

- **Static**
- **Access** protected

Constructor *void* function AS2Exception::__construct([\$message = "], [\$code = self::DEFAULT_ERROR], [\$previous = null]) [*line 65*]

Function Parameters:

- **\$message**
- **\$code**
- **\$previous**

- **Access** public

void function AS2Exception::getLevel() [*line 74*]

- **Access** public

void function AS2Exception::getMessageShort() [*line 82*]

- **Access** public

Class AS2Header

[*line 32*]

AS2Secure - PHP Lib for AS2 message encoding / decoding

- **Package** AS2Secure
- **Author** Sebastien MALOT < contact@as2secure.com >
- **Version** 0.8.4
- **Copyright** Copyright (c) 2010, Sebastien MALOT

Last release at : <http://www.as2secure.com>

This file is part of AS2Secure Project.

AS2Secure is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the

License, or (at your option) any later version.

AS2Secure is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with AS2Secure.

- **License** [GNU General Public License](#)

AS2Header::\$headers

mixed = array() [line 33]

- **Access** protected

AS2Header::\$_position

mixed = null [line 35]

- **Access** protected

Constructor *void* function AS2Header::__construct([\$data = null]) [line 37]

Function Parameters:

- **\$data**

- **Access** public

void function AS2Header::addHeader(\$key, \$value) [line 62]

Function Parameters:

- *string* **\$key** The name of the header
- *string* **\$value** The value of the header

Add new header (or override current one)

- **Access** public

void function AS2Header::addHeaders(\$values, \$headers) [*line 71*]

Function Parameters:

- *array* **\$headers** The new headers to use
- **\$values**

Add a set of headers (or override currents)

- **Access** public

void function AS2Header::addHeadersFromMessage(\$message) [*line 81*]

Function Parameters:

- *string* **\$message** The message content to use

Add a set of headers extracted from a mime message

- **Access** public

int function AS2Header::count() [*line 139*]

Return the count of headers

- **Access public**

void function AS2Header::current() [*line 194*]

- **Access public**

boolean function AS2Header::exists(\$key) [*line 150*]

Function Parameters:

- *string* **\$key** The header to check existence

Check if an header exists

- **Access public**

string function AS2Header::getHeader(\$key) [*line 127*]

Function Parameters:

- *string* **\$key** The header

Return the value of an header

- **Access** public

array function AS2Header::getHeaders() [*line 103*]

Return all headers as an array

- **Access** public

void function AS2Header::key() [*line 198*]

- **Access** public

void function AS2Header::next() [*line 203*]

- **Access** public

void function AS2Header::offsetExists(\$offset) [*line 174*]

Function Parameters:

- **\$offset**

- **Access** public

void function AS2Header::offsetGet(\$offset) [*line 178*]

Function Parameters:

- **\$offset**

- **Access public**

void function AS2Header::offsetSet(\$offset, \$value) [line 182]

Function Parameters:

- **\$offset**
- **\$value**

- **Access public**

void function AS2Header::offsetUnset(\$offset) [line 186]

Function Parameters:

- **\$offset**

- **Access public**

object AS2Header function AS2Header::parseHttpRequest(0) [line 247]

Function Parameters:

- *string* **0** The content to parse

Extract headers from http request and return a new instance of AS2Header

- **Static**
- **Access** public

object AS2Header function AS2Header::parseText(\$text) [*line 222*]

Function Parameters:

- *string* **\$text** The content to parse

Extract headers from mime message and return a new instance of AS2Header

- **Static**
- **Access** public

void function AS2Header::removeHeader(\$key) [*line 94*]

Function Parameters:

- *string* **\$key** The name of the header

Remove an header

- **Access** public

void function AS2Header::rewind() [*line 207*]

- **Access** public

void function AS2Header::setHeaders(\$headers) [*line 51*]

Function Parameters:

- *array* **\$headers** The new headers to use

Reset all current headers with new values

- **Access** public

array function AS2Header::toFormattedArray() [*line 112*]

Return all headers as a formatted array

- **Access** public

void function AS2Header::valid() [*line 211*]

- **Access** public

string function AS2Header::__toString() [*line 160*]

Magic method that returns headers serialized as in mime message

- **Access** public

Class AS2Log

[line 32]

AS2Secure - PHP Lib for AS2 message encoding / decoding

- **Package** AS2Secure
- **Author** Sebastien MALOT < contact@as2secure.com >
- **Version** 0.8.4
- **Copyright** Copyright (c) 2010, Sebastien MALOT

Last release at : <http://www.as2secure.com>

This file is part of AS2Secure Project.

AS2Secure is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

AS2Secure is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with AS2Secure.

- **License** [GNU General Public License](#)

AS2Log::ERROR

= 'error' *[line 38]*

AS2Log::FAILURE

= 'failure' *[line 40]*

AS2Log::INFO

= 'info' *[line 37]*

AS2Log::WARNING

= 'warning' *[line 39]*

AS2Log::\$code

mixed = 0 *[line 47]*

- **Access** protected

AS2Log::\$current_message_id

mixed = " [line 43]

- **Static**
- **Access** protected

AS2Log::\$filename

mixed = 'events.log' [line 33]

- **Static**
- **Access** public

AS2Log::\$level

mixed = self::INFO [line 48]

- **Access** protected

AS2Log::\$message

mixed = " [line 46]

- **Access** protected

AS2Log::\$message_id

mixed = " [line 45]

- **Access** protected

AS2Log::\$stack

mixed = array() [line 35]

- **Static**
- **Access** protected

Constructor *void* function AS2Log::__construct(\$message, [\$code = 0], [\$level = self::INFO]) [line 55]

Function Parameters:

- **\$message**
- **\$code**
- **\$level**

Constructor

- **Access** protected

void function AS2Log::error(\$message_id, \$message, [\$code = 0]) [line 147]

Function Parameters:

- *string* **\$message_id** The current message id (from mime message)
- *string* **\$message** The message to log
- *int* **\$code** The code number

Static handler for error level

- **Static**
- **Access** public

int function AS2Log::getCode() [*line 105*]

Return the code

- **Access** public

array function AS2Log::getCount([\$level = null]) [*line 191*]

Function Parameters:

- *string* **\$level** The level to search for

Return the stack of log for a specific level (or not)

- **Static**
- **Access** public

string function AS2Log::getLastLogEvents([\$count = 40], [\$reverse = true]) [*line 215*]

Function Parameters:

- *int* **\$count** The count of log lines
- *boolean* **\$reverse** To reverse order of log lines

Return the lasts logs from logfile (work only on unix systems)

- **Static**
- **Access** public

string function AS2Log::getLevel() [*line 114*]

Return the level

- **Access** public

string function AS2Log::getMessage() [*line 96*]

Return the message

- **Access** public

string function AS2Log::getMessageId() [*line 87*]

Return the message id (from mime part)

- **Access** public

array function AS2Log::getStack([\$level = null]) [*line 173*]

Function Parameters:

- *string* **\$level** The level to search for

Return the stack of log for a specific level (or not)

- **Static**
- **Access** public

void function AS2Log::handleLevel(\$level, \$message_id, \$message, [\$code = 0]) [*line 159*]

Function Parameters:

- *string* **\$level** The level
- *string* **\$message_id** The current message id (from mime message)
- *string* **\$message** The message to log
- *int* **\$code** The code number

Static and generic handler

- **Static**
- **Access** public

boolean function AS2Log::hasError() [*line 200*]

Indicate if there is errors into the stack

- **Static**
- **Access** public

void function AS2Log::info(\$message_id, \$message, [\$code = 0]) [*line 125*]

Function Parameters:

- *string* **\$message_id** The current message id (from mime message)
- *string* **\$message** The message to log
- *int* **\$code** The code number

Static handler for info level

- **Static**
- **Access** public

void function AS2Log::logEvent(\$event, [\$log_message_id = true]) [*line 69*]

Function Parameters:

- *object* **\$event** The event to log into file
- **\$log_message_id**

Log event into a file

- **Access** protected

void function AS2Log::warning(\$message_id, \$message, [\$code = 0]) [*line 136*]

Function Parameters:

- *string* **\$message_id** The current message id (from mime message)
- *string* **\$message** The message to log
- *int* **\$code** The code number

Static handler for warning level

- **Static**
- **Access** public

Class AS2MDN

[line 32]

AS2Secure - PHP Lib for AS2 message encoding / decoding

- **Package** AS2Secure
- **Author** Sebastien MALOT < contact@as2secure.com>
- **Version** 0.8.4
- **Copyright** Copyright (c) 2010, Sebastien MALOT

Last release at : <http://www.as2secure.com>

This file is part of AS2Secure Project.

AS2Secure is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

AS2Secure is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with AS2Secure.

- **License** [GNU General Public License](#)

AS2MDN::ACTION_AUTO

= 'automatic-action' *[line 57]*

Refers to RFC 4130

<http://rfclibrary.hosting.com/rfc/rfc4130/rfc4130-34.asp>

AS2MDN::ACTION_MANUAL

= 'manual-action' *[line 58]*

AS2MDN::MODIFIER_ERROR

= 'error' *[line 66]*

AS2MDN::MODIFIER_WARNING

= 'warning' *[line 67]*

AS2MDN::SENDING_AUTO

= 'MDN-sent-automatically' [*line 60*]

AS2MDN::SENDING_MANUAL

= 'MDN-sent-manually' [*line 61*]

AS2MDN::TYPE_FAILED

= 'failed' [*line 64*]

AS2MDN::TYPE_PROCESSED

= 'processed' [*line 63*]

AS2MDN::\$attributes

mixed = null [*line 51*]

Valid tokens :

action-mode	:	"manual-action" "automatic-action"	
sending-mode	:	"MDN-sent-manually" "MDN-sent-automatically"	
disposition-type	:	"processed" "failed"	
disposition-modifier	:	("error" "warning")	disposition-modifier-extension
disposition-modifier-extension	:	(cf : AS2Exception values)	encoded-
message-digest	:	(base64 format + digest- <i>alg-id</i> = "sha1" "md5")	
reporting-ua	:	user-agent	

- **Access** protected

AS2MDN::\$message

mixed = " [*line 36*]

Human readable message

- **Access** protected

AS2MDN::\$url

mixed = " [line 38]

- **Access** protected

Constructor *void* function AS2MDN::__construct([\$data = null], [\$params = array()]) *[line 70]*

Function Parameters:

- **\$data**
- **\$params**

- **Access** public

void function AS2MDN::decode() *[line 272]*

Decode MDN stored into path file and set attributes

- **Access** public

void function AS2MDN::encode([\$message = null]) *[line 185]*

Function Parameters:

- *object* **\$message** The referring message

Encode and generate MDN from attributes and message (if exists)

- **Access** public

string function AS2MDN::getAttribute(\$key) [*line 167*]

Function Parameters:

- *string* **\$key** Token

Return an attribute from computer readable message

- **Access** public

array function AS2MDN::getAttributes() [*line 176*]

Return the computer readable message

- **Access** public

string function AS2MDN::getMessage() [*line 146*]

Return the human readable message

- **Access** public

string function AS2MDN::getUrl() [*line 311*]

Return the url to send message

- **Access** public

void function AS2MDN::setAttribute(\$key, \$value) [*line 156*]

Function Parameters:

- *string* **\$key** Token
- *string* **\$value** Value

Set attribute for computer readable message

- **Access** public

void function AS2MDN::setMessage(\$message) [*line 137*]

Function Parameters:

- *string* **\$message** The message to set

Set the human readable message

- **Access** public

string function AS2MDN::__toString() [*line 128*]

Return the human readable message

- **Access** public

Class AS2Message

[line 32]

AS2Secure - PHP Lib for AS2 message encoding / decoding

- **Package** AS2Secure
- **Author** Sebastien MALOT < contact@as2secure.com >
- **Version** 0.8.4
- **Copyright** Copyright (c) 2010, Sebastien MALOT

Last release at : <http://www.as2secure.com>

This file is part of AS2Secure Project.

AS2Secure is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

AS2Secure is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with AS2Secure.

- **License** [GNU General Public License](#)

AS2Message::\$mic_checksum

mixed = false [line 34]

- **Access** protected

Constructor *void* function AS2Message::__construct(\$data, [\$params = array()]) [line 36]

Function Parameters:

- **\$data**
- **\$params**

- **Access** public

boolean function AS2Message::addFile(\$data, [\$mimetype = ""], [\$filename = ""], [\$is_file = true], [\$encoding = 'binary']) [*line 68*]

Function Parameters:

- *string* **\$data** The content or the file
- *string* **\$mimetype** The mimetype of the message
- *boolean* **\$is_file** If file
- *string* **\$encoding** The encoding to use for transfert
- **\$filename**

Add file to the message

- **Access** public

array function AS2Message::decode() [*line 248*]

Decode message extracting files from message

- **Access** public

void function AS2Message::encode() [*line 130*]

Build message and encode it (signing and/or crypting)

- **Access** public

object The function AS2Message::generateMDN([\$exception = null]) [*line 261*]

Function Parameters:

- *object* **\$exception** The exception if error handled

Generate a MDN from the message

- **Access** public

array function AS2Message::getAuthentication() [*line 120*]

Return the authentication to use to send message to the partner

- **Access** public

array function AS2Message::getFiles() [*line 93*]

Return files which compose the message (should contain at least one file)

- **Access** public

string function AS2Message::getMicChecksum() [*line 102*]

Return the last calculated checksum

- **Access** public

string function AS2Message::getUrl() [*line 111*]

Return the url to send message

- **Access** public

Class AS2Partner

[*line 32*]

AS2Secure - PHP Lib for AS2 message encoding / decoding

- **Package** AS2Secure
- **Author** Sebastien MALOT < contact@as2secure.com >
- **Version** 0.8.4
- **Copyright** Copyright (c) 2010, Sebastien MALOT

Last release at : <http://www.as2secure.com>

This file is part of AS2Secure Project.

AS2Secure is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

AS2Secure is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with AS2Secure.

- **License** [GNU General Public License](#)

AS2Partner::ACK_ASYNC

= 'ASYNC' [*line 88*]

AS2Partner::ACK_SYNC

= 'SYNC' *[line 87]*

AS2Partner::CRYPT_3DES

= 'des3' *[line 101]*

AS2Partner::CRYPT_AES_128

= 'aes128' *[line 102]*

AS2Partner::CRYPT_AES_192

= 'aes192' *[line 103]*

AS2Partner::CRYPT_AES_256

= 'aes256' *[line 104]*

AS2Partner::CRYPT_DES

= 'des' *[line 100]*

AS2Partner::CRYPT_NONE

= 'none' *[line 96]*

AS2Partner::CRYPT_RC2_40

= 'rc2-40' *[line 97]*

AS2Partner::CRYPT_RC2_64

= 'rc2-64' *[line 98]*

AS2Partner::CRYPT_RC2_128

= 'rc2-128' *[line 99]*

AS2Partner::ENCODING_BASE64

= 'base64' *[line 83]*

AS2Partner::ENCODING_BINARY

= 'binary' *[line 84]*

AS2Partner::METHOD_AUTO

= CURLAUTH_ANY *[line 76]*

AS2Partner::METHOD_BASIC

= CURLAUTH_BASIC *[line 77]*

AS2Partner::METHOD_DIGEST

= CURLAUTH_DIGEST *[line 78]*

AS2Partner::METHOD_GSS

= CURLAUTH_GSSNEGOTIATE [*line 80*]

AS2Partner::METHOD_NONE

= 'NONE' [*line 75*]

AS2Partner::METHOD_NTLM

= CURLAUTH_NTLM [*line 79*]

AS2Partner::SIGN_MD5

= 'md5' [*line 93*]

AS2Partner::SIGN_NONE

= 'none' [*line 91*]

AS2Partner::SIGN_SHA1

= 'sha1' [*line 92*]

AS2Partner::\$comment

mixed = " [*line 38*]

- **Access** protected

AS2Partner::\$connector_class

mixed = 'AS2Connector' [*line 69*]

- **Access** protected

AS2Partner::\$email

mixed = " [*line 37*]

- **Access** protected

AS2Partner::\$id

mixed = " [line 36]

- **Access** protected

AS2Partner::\$is_local

mixed = false [line 34]

- **Access** protected

AS2Partner::\$mdn_credencial_login

mixed = " [line 65]

- **Access** protected

AS2Partner::\$mdn_credencial_method

mixed = self::METHOD_NONE [line 64]

- **Access** protected

AS2Partner::\$mdn_credencial_password

mixed = " [line 66]

- **Access** protected

AS2Partner::\$mdn_request

mixed = self::ACK_SYNC [*line 62*]

- **Access** protected

AS2Partner::\$mdn_signed

mixed = true [*line 63*]

- **Access** protected

AS2Partner::\$mdn_subject

mixed = 'AS2 MDN Subject' [*line 61*]

- **Access** protected

AS2Partner::\$mdn_url

mixed = " [*line 60*]

- **Access** protected

AS2Partner::\$name

mixed = " [*line 35*]

- **Access** protected

AS2Partner::\$sec_certificate

mixed = " [line 44]

- **Access** protected

AS2Partner::\$sec_encrypt_algorithm

mixed = self::CRYPT_3DES [line 47]

- **Access** protected

AS2Partner::\$sec_pkcs12

mixed = " [line 41]

- **Access** protected

AS2Partner::\$sec_pkcs12_password

mixed = " [line 42]

- **Access** protected

AS2Partner::\$sec_signature_algorithm

mixed = self::SIGN_SHA1 [line 46]

- **Access** protected

AS2Partner::\$send_compress

mixed = false [line 50]

- **Access** protected

AS2Partner::\$send_content_type

mixed = 'application/EDI-Consent' [line 53]

- **Access** protected

AS2Partner::\$send_credencial_login

mixed = " [line 55]

- **Access** protected

AS2Partner::\$send_credencial_method

mixed = self::METHOD_NONE [line 54]

- **Access** protected

AS2Partner::\$send_credencial_password

mixed = " [line 56]

- **Access** protected

AS2Partner::\$send_encoding

mixed = self::ENCODING_BASE64 [*line 57*]

- **Access** protected

AS2Partner::\$send_subject

mixed = 'AS2 Message Subject' [*line 52*]

- **Access** protected

AS2Partner::\$send_url

mixed = " [*line 51*]

- **Access** protected

AS2Partner::\$stack

mixed = array() [*line 72*]

- **Static**
- **Access** protected

Constructor *void* function AS2Partner::__construct(\$data) [*line 184*]

Function Parameters:

- *data* **\$data** The data to set from

Restricted constructor

- **Access** protected

array function AS2Partner::getAvailablesEncryptions() [*line 123*]

Return the list of available cypher

- **Static**
- **Access** public

array function AS2Partner::getAvailablesSignatures() [*line 111*]

Return the list of available signatures

- **Static**
- **Access** public

object : function AS2Partner::getPartner(\$partner_id, [\$reload = false]) [*line 145*]

Function Parameters:

- *partner_id* **\$partner_id** String : Partner ID (case sensitive) corresponds to AS2-To / AS2-From headers
- *reload* **\$reload** Boolean : Allow to reload config from file

Return an AS2Partner object for a specified Partner ID

- **Static**
- **Access** public

Return function AS2Partner::__get(\$key) [line 202]

Function Parameters:

- *key* **\$key** Property name

Magic getter

- **Access** public

void function AS2Partner::__set(\$key, \$value) [line 216]

Function Parameters:

- *key* **\$key** Property name
- *value* **\$value** New value to set

Magic setter

- **Access** public

void function AS2Partner::__toString() [line 226]

Magic method

- **Access** public

Class AS2Request

[line 32]

AS2Secure - PHP Lib for AS2 message encoding / decoding

- **Package** AS2Secure
- **Author** Sebastien MALOT < contact@as2secure.com >
- **Version** 0.8.2
- **Copyright** Copyright (c) 2010, Sebastien MALOT

Last release at : <http://www.as2secure.com>

This file is part of AS2Secure Project.

AS2Secure is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

AS2Secure is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with AS2Secure.

- **License** [GNU General Public License](#)

AS2Request::\$request

mixed = null [line 33]

- **Access** protected

Constructor *void* function AS2Request::__construct(\$content, \$headers) [line 35]

Function Parameters:

- **\$content**
- **\$headers**

- **Access public**

void function AS2Request::decode() [line 224]

- **Access public**

void function AS2Request::decrypt() [line 58]

- **Access public**

void function AS2Request::encode() [line 220]

- **Access public**

void function AS2Request::getObject() [line 86]

- **Access public**

Class AS2Server

[line 33]

AS2Secure - PHP Lib for AS2 message encoding / decoding

- **Package** AS2Secure
- **Author** Sebastien MALOT < contact@as2secure.com>
- **Version** 0.8.4
- **Copyright** Copyright (c) 2010, Sebastien MALOT

Last release at : <http://www.as2secure.com>

This file is part of AS2Secure Project.

AS2Secure is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or (at your option) any later version.

AS2Secure is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License for more details.

You should have received a copy of the GNU General Public License along with AS2Secure.

- **License** [GNU General Public License](#)

AS2Server::TYPE_MDN

= 'MDN' [*line 35*]

AS2Server::TYPE_MESSAGE

= 'Message' [*line 34*]

void function AS2Server::closeConnectionAndWait(\$sleep) [*line 270*]

Function Parameters:

- *int* **\$sleep** The number of seconds to wait for

Close current HTTP connection and wait some secons

- **Static**
- **Access** protected

request function AS2Server::handle([\$request = null]) [*line 44*]

Function Parameters:

- *request* **\$request** (If not set, get data from standard input)

Handle a request (server side)

- **Static**
- **Access** public

String function AS2Server::saveMessage(\$content, \$headers, [\$filename = ''], [\$type = 'raw']) [*line 236*]

Function Parameters:

- *content* **\$content** The content to save (mandatory)
- *headers* **\$headers** The headers to save (optional)
- *filename* **\$filename** The filename to use if known (optional)
- *type* **\$type** Values : raw | decrypted | payload (mandatory)

Save the content of the request for futur handle and/or backup

- **Static**
- **Access** protected

Package Horde_Util Classes

Class Horde_String

[line 18]

The `Horde_String::` class provides static methods for charset and locale safe string manipulation.

\$Horde: framework/Util/Horde_String.php,v 1.43.6.37 2009/03/30 15:31:38 jan Exp \$

Copyright 2003-2009 The Horde Project (<http://www.horde.org/>)

See the enclosed file COPYING for license information (LGPL). If you did not receive this file, see <http://www.fsf.org/copyleft/lgpl.html>.

- **Package** Horde_Util
- **Author** Jan Schneider < jan@horde.org >
- **Since** Horde 3.0

Horde_String::\$charset

mixed = 'UTF-8' [line 20]

- **Static**
- **Access** public

mixed function Horde_String::convertCharset(\$input, \$from, [\$to = null]) *[line 76]*

Function Parameters:

- *mixed* **\$input** The data to be converted. If \$input is an an array, the array's values get converted recursively.
- *string* **\$from** The string's current charset.
- *string* **\$to** The charset to convert the string to. If not specified, the global variable `$_HORDE_STRING_CHARSET` will be used.

Converts a string from one charset to another.

Works only if either the iconv or the mbstring extension are present and best if both are available. The original string is returned if conversion failed or none of the extensions were available.

- **Access** public

boolean function Horde_String::extensionExists(\$ext) [*line 31*]

Function Parameters:

- *string* **\$ext** The extension name.

Caches the result of extension_loaded() calls.

- **See** Util::extensionExists()
- **Access** public

boolean function Horde_String::isAlpha(\$string, [\$charset = null]) [*line 586*]

Function Parameters:

- *\$string* **\$string** The string to test.
- *\$charset* **\$charset** The charset to use when testing the string.

Returns true if the every character in the parameter is an alphabetic character.

- **Access public**

boolean function Horde_String::isLower(\$string, [\$charset = null]) [line 618]

Function Parameters:

- *\$string* **\$string** The string to test.
- *\$charset* **\$charset** The charset to use when testing the string.

Returns true if ever character in the parameter is a lowercase letter in the current locale.

- **Access public**

boolean function Horde_String::isUpper(\$string, [\$charset = null]) [line 633]

Function Parameters:

- *string* **\$string** The string to test.
- *string* **\$charset** The charset to use when testing the string.

Returns true if every character in the parameter is an uppercase letter in the current locale.

- **Access public**

string function Horde_String::length(\$string, [\$charset = null]) [line 359]

Function Parameters:

- *string* **\$string** The string to return the length of.
- *string* **\$charset** The charset to use when calculating the string's length.

Returns the character (not byte) length of a string.

- **Access** public

string function Horde_String::lower(\$string, [\$locale = false], [\$charset = null]) [*line 193*]

Function Parameters:

- *string* **\$string** The string to be converted.
- *boolean* **\$locale** If true the string will be converted based on a given charset, locale independent else.
- *string* **\$charset** If \$locale is true, the charset to use when converting. If not provided the current charset.

Makes a string lowercase.

- **Access** public

string function Horde_String::pad(\$input, \$length, [\$pad = ' '], [\$type = STR_PAD_RIGHT], [\$charset = null]) [*line 426*]

Function Parameters:

- *string* **\$input** The string to be padded.
- *integer* **\$length** The length of the resulting string.
- *string* **\$pad** The string to pad the input string with. Must be in the same charset like the input string.
- *const* **\$type** The padding type. One of STR_PAD_LEFT, STR_PAD_RIGHT, or STR_PAD_BOTH.
- *string* **\$charset** The charset of the input and the padding strings.

Returns a string padded to a certain length with another string.

This method behaves exactly like `str_pad` but is multibyte safe.

- **Access** public

integer function Horde_String::pos(\$haystack, \$needle, [\$offset = 0], [\$charset = null]) [*line 392*]

Function Parameters:

- *string* **\$haystack** The string to search through.
- *string* **\$needle** The string to search for.
- *integer* **\$offset** Allows to specify which character in haystack to start searching.
- *string* **\$charset** The charset to use when searching for the \$needle string.

Returns the numeric position of the first occurrence of \$needle in the \$haystack string.

- **Access** public

array function Horde_String::regexMatch(\$text, \$regex, [\$charset = null]) [*line 651*]

Function Parameters:

- *string* **\$text** The text to search.
- *array* **\$regex** The regular expressions to use, without perl regex delimiters (e.g. '/' or '|').
- *string* **\$charset** The character set of the text.

Performs a multibyte safe regex match search on the text provided.

- **Since** Horde 3.1
- **Access** public

void function Horde_String::setDefaultCharset(\$charset) [line 48]

Function Parameters:

- *string* **\$charset** The charset to use as the default one.

Sets a default charset that the Horde_String:: methods will use if none is explicitly specified.

- **Access** public

string function Horde_String::substr(\$string, \$start, [\$length = null], [\$charset = null]) [line 308]

Function Parameters:

- *string* **\$string** The string to be converted.
- *integer* **\$start** The part's start position, zero based.
- *integer* **\$length** The part's length.
- *string* **\$charset** The charset to use when calculating the part's length, defaults to current charset. position and

Returns part of a string.

- **Access** public

string function Horde_String::ucfirst(\$string, [\$locale = false], [\$charset = null]) [line 283]

Function Parameters:

- *string* **\$string** The string to be capitalized.
- *boolean* **\$locale** If true the string will be converted based on a locale independent else. given charset,
- *string* **\$charset** The charset to use, defaults to current charset.

Returns a string with the first letter capitalized if it is alphabetic.

- **Access** public

string function Horde_String::upper(\$string, [\$locale = false], [\$charset = null]) [*line 238*]

Function Parameters:

- *string* **\$string** The string to be converted.
- *boolean* **\$locale** If true the string will be converted based on a given charset, locale independent else.
- *string* **\$charset** If \$locale is true, the charset to use when converting. If not provided the current charset.

Makes a string uppercase.

- **Access** public

string function Horde_String::wordwrap(\$string, [\$width = 75], [\$break = "\n"], [\$cut = false], [\$charset = null], [\$line_folding = false]) [*line 483*]

Function Parameters:

- *string* **\$string** Horde_String containing the text to wrap.
- *integer* **\$width** Wrap the string at this number of characters.
- *string* **\$break** Character(s) to use when breaking lines.
- *boolean* **\$cut** Whether to cut inside words if a line can't be wrapped.
- *string* **\$charset** Character set to use when breaking lines.
- *boolean* **\$line_folding** Whether to apply line folding rules per RFC 822 or similar. The correct break characters including leading whitespace have to be specified too.

Wraps the text of a message.

- **Since** Horde 3.2
- **Access** public

string function Horde_String::wrap(\$text, [\$length = 80], [\$break_char = "\n"], [\$charset = null], [\$quote = false]) [*line 553*]

Function Parameters:

- *string* **\$text** Horde_String containing the text to wrap.
- *integer* **\$length** Wrap \$text at this number of characters.
- *string* **\$break_char** Character(s) to use when breaking lines.
- *string* **\$charset** Character set to use when breaking lines.
- *boolean* **\$quote** Ignore lines that are wrapped with the '>' character (RFC 2646)? If true, we don't remove any padding whitespace at the end of the string.

Wraps the text of a message.

- **Access** public

string function Horde_String::_convertCharset(\$input, \$from, \$to) [*line 129*]

Function Parameters:

- *string* **\$input** See Horde_String::convertCharset().
- *string* **\$from** See Horde_String::convertCharset().
- *string* **\$to** See Horde_String::convertCharset().

Internal public function used to do charset conversion.

- **Access** protected

string function Horde_String::_mbstringCharset(\$charset) [*line 681*]

Function Parameters:

- *string* **\$charset** The original charset.

Workaround charsets that don't work with mbstring public functions.

- **Access** protected

Appendices

Appendix A - Class Trees

Package Horde_MIME

Horde_MIME

- [Horde_MIME](#)

Horde_MIME_Part

- [Horde_MIME_Part](#)
 - [Horde_MIME_Message](#)

Horde_MIME_Structure

- [Horde_MIME_Structure](#)

Package AS2Secure

AS2Abstract

- [AS2Abstract](#)
 - [AS2MDN](#)
 - [AS2Message](#)
 - [AS2Request](#)

AS2Adapter

- [AS2Adapter](#)

AS2Client

- [AS2Client](#)

AS2Connector

- [AS2Connector](#)

AS2Exception

- Exception
 - [AS2Exception](#)

AS2Header

- [AS2Header](#)

AS2Log

- [AS2Log](#)

AS2Partner

- [AS2Partner](#)

AS2Server

- [AS2Server](#)

Package Horde_Util

Horde_String

- [Horde_String](#)

Package Horde_Horde_Util

Horde_Util

- [Horde_Util](#)

Index

A

AS2MDN:: toString()	97
<i>Return the human readable message</i>	
AS2MDN::setMessage()	97
<i>Set the human readable message</i>	
AS2Message	98
<i>AS2Secure - PHP Lib for AS2 message encoding / decoding</i>	
AS2Message::\$mic_checksum	98
AS2Message::decode()	99
<i>Decode message extracting files from message</i>	
AS2Message::addFile()	99
<i>Add file to the message</i>	
AS2MDN::setAttribute()	97
<i>Set attribute for computer readable message</i>	
AS2MDN::getUrl()	96
<i>Return the url to send message</i>	
AS2MDN::encode()	95
<i>Encode and generate MDN from attributes and message (if exists)</i>	
AS2MDN::decode()	95
<i>Decode MDN stored into path file and set attributes</i>	
AS2MDN::getAttribute()	96
<i>Return an attribute from computer readable message</i>	
AS2MDN::getAttributes()	96
<i>Return the computer readable message</i>	
AS2MDN::getMessage()	96
<i>Return the human readable message</i>	
AS2Message::encode()	99
<i>Build message and encode it (signing and/or crypting)</i>	
AS2Message::generateMDN()	100
<i>Generate a MDN from the message</i>	
AS2Partner::CRYPT_AES_128	102
AS2Partner::CRYPT_3DES	102
AS2Partner::CRYPT_AES_192	102
AS2Partner::CRYPT_AES_256	102
AS2Partner::CRYPT_NONE	102
AS2Partner::CRYPT_DES	102
AS2Partner::ACK_SYNC	102
AS2Partner::ACK_ASYNC	101
AS2Message::getFiles()	100
<i>Return files which compose the message (should contain at least one file)</i>	
AS2Message::getAuthentication()	100
<i>Return the authentication to use to send message to the partner</i>	
AS2Message::getMicChecksum()	100
<i>Return the last calculated checksum</i>	
AS2Message::getUrl()	101

Return the url to send message	101
AS2Partner	101
AS2Secure - PHP Lib for AS2 message encoding / decoding	
AS2MDN::\$url	95
AS2MDN::\$message	94
Human readable message	
AS2Log::getLevel()	90
Return the level	
AS2Log::getLastLogEvents()	89
Return the lasts logs from logfile (work only on unix systems)	
AS2Log::getMessage()	90
Return the message	
AS2Log::getMessageId()	90
Return the message id (from mime part)	
AS2Log::handleLevel()	91
Static and generic handler	
AS2Log::getStack()	90
Return the stack of log for a specific level (or not)	
AS2Log::getCount()	89
Return the stack of log for a specific level (or not)	
AS2Log::getCode()	89
Return the code	
AS2Log::\$message	87
AS2Log::\$level	87
AS2Log::\$message_id	88
AS2Log::\$stack	88
AS2Log::error()	88
Static handler for error level	
AS2Log::hasError()	91
Indicate if there is errors into the stack	
AS2Log::info()	91
Static handler for info level	
AS2MDN::SENDING_MANUAL	94
AS2MDN::SENDING_AUTO	94
AS2MDN::TYPE_FAILED	94
AS2MDN::TYPE_PROCESSED	94
AS2MDN::\$attributes	94
Valid tokens :	
AS2MDN::MODIFIER_WARNING	93
AS2MDN::MODIFIER_ERROR	93
AS2Log::warning()	92
Static handler for warning level	
AS2Log::logEvent()	92
Log event into a file	
AS2MDN	93
AS2Secure - PHP Lib for AS2 message encoding / decoding	
AS2MDN::ACTION_AUTO	93
Refers to RFC 4130	
AS2MDN::ACTION_MANUAL	93
AS2Partner::CRYPT_RC2_40	102
AS2Partner::CRYPT_RC2_64	102
AS2Partner::\$send_url	108
AS2Partner::\$send_subject	108

AS2Partner::\$stack	108
AS2Partner::getAvailablesEncryptions()	109
<i>Return the list of available cypher</i>	
AS2Partner::getPartner()	109
<i>Return an AS2Partner object for a specified Partner ID</i>	
AS2Partner::getAvailablesSignatures()	109
<i>Return the list of available signatures</i>	
AS2Partner::\$send_encoding	108
AS2Partner::\$send_credencial_password	107
AS2Partner::\$send_compress	107
AS2Partner::\$sec_signature_algorithm	106
AS2Partner::\$send_content_type	107
AS2Partner::\$send_credencial_login	107
AS2Partner::\$send_credencial_method	107
AS2Partner::_get()	110
<i>Magic getter</i>	
AS2Partner::_set()	110
<i>Magic setter</i>	
AS2Server::TYPE_MDN	113
AS2Server	112
<i>AS2Secure - PHP Lib for AS2 message encoding / decoding</i>	
AS2Server::TYPE_MESSAGE	113
AS2Server::closeConnectionAndWait()	113
<i>Close current HTTP connection and wait some secs</i>	
AS2Server::saveMessage()	114
<i>Save the content of the request for futur handle and/or backup</i>	
AS2Server::handle()	114
<i>Handle a request (server side)</i>	
AS2Request::getObject()	112
AS2Request::encode()	112
AS2Request	111
<i>AS2Secure - PHP Lib for AS2 message encoding / decoding</i>	
AS2Partner::_toString()	110
<i>Magic method</i>	
AS2Request::\$request	111
AS2Request::decode()	112
AS2Request::decrypt()	112
AS2Partner::\$sec_pkcs12_password	106
AS2Partner::\$sec_pkcs12	106
AS2Partner::METHOD_NTLM	103
AS2Partner::METHOD_NONE	103
AS2Partner::SIGN_MD5	103
AS2Partner::SIGN_NONE	103
AS2Partner::\$comment	103
AS2Partner::SIGN_SHA1	103
AS2Partner::METHOD_GSS	103
AS2Partner::METHOD_DIGEST	102
AS2Partner::ENCODING_BASE64	102
AS2Partner::CRYPT_RC2_128	102
AS2Partner::ENCODING_BINARY	102
AS2Partner::METHOD_AUTO	102
AS2Partner::METHOD_BASIC	102
AS2Partner::\$connector_class	103

AS2Partner::\$email	103
AS2Partner::\$mdn_url	105
AS2Partner::\$mdn_subject	105
AS2Partner::\$name	105
AS2Partner::\$sec_certificate	106
AS2Partner::\$sec_encrypt_algorithm	106
AS2Partner::\$mdn_signed	105
AS2Partner::\$mdn_request	105
AS2Partner::\$is_local	104
AS2Partner::\$id	104
AS2Partner::\$mdn_credencial_login	104
AS2Partner::\$mdn_credencial_method	104
AS2Partner::\$mdn_credencial_password	104
AS2Log::\$filename	87
AS2Log::\$current_message_id	87
AS2Adapter::\$ssl_openssl	63
AS2Adapter::\$ssl_adapter	62
<i>Allow to specify full path to main applications</i>	
AS2Adapter::\$tmp_files	63
<i>Array to store temporary files created and scheduled to unlink</i>	
AS2Adapter::addTempFileForDelete()	64
<i>Schedule file for deletion</i>	
AS2Adapter::compose()	64
<i>Generate a mime multipart file from files</i>	
AS2Adapter::calculateMicChecksum()	64
<i>Calculate the message integrity check (MIC) using SHA1 or MD5 algo</i>	
AS2Adapter::\$partner_to	62
AS2Adapter::\$partner_from	62
AS2Abstract::setPartnerFrom()	61
AS2Abstract::setMessageId()	60
AS2Abstract::setPartnerTo()	61
AS2Adapter	61
<i>AS2Secure - PHP Lib for AS2 message encoding / decoding</i>	
AS2Adapter::\$javapath	62
AS2Adapter::compress()	65
<i>Compress a file which contains mime headers</i>	
AS2Adapter::decompress()	65
<i>Decompress a file which contains mime headers</i>	
AS2Adapter::getMicChecksum()	68
<i>Extract the message integrity check (MIC) from the digital signature</i>	
AS2Adapter::getDataFromPKCS12()	68
<i>Extract Part from a PKCS12 Certificate</i>	
AS2Adapter::getPrivateFromPKCS12()	68
<i>Extract Private Certificate from a PKCS12 Certificate</i>	
AS2Adapter::getPublicFromPKCS12()	69
<i>Extract Public certificate from a PKCS12 Certificate</i>	
AS2Adapter::hex2bin()	69
<i>Convert a string from hexadecimal format to binary format</i>	
AS2Adapter::getTempFilename()	69
<i>Create a temporary file into temporary directory and add it to the garbage collector at shutdown</i>	
AS2Adapter::getCAFromPKCS12()	67
<i>Extract CA from a PKCS12 Certificate</i>	

AS2Adapter::extract()	67
<i>Extract files from mime multipart file</i>	
AS2Adapter::detectMimeType()	66
<i>Determine the mimetype of a file (also called 'Content-Type')</i>	
AS2Adapter::decrypt()	65
<i>Decrypt a message</i>	
AS2Adapter::detectOS()	66
<i>Determine the Server OS</i>	
AS2Adapter::encrypt()	66
<i>Encrypt a file</i>	
AS2Adapter::exec()	67
<i>Execute a command line and throw Exception if an error appends</i>	
AS2Abstract::setHeaders()	60
AS2Abstract::isSigned()	60
AS2Abstract::\$partner_from	56
AS2Abstract::\$mimetype	56
AS2Abstract::\$partner_to	56
AS2Abstract::\$path	56
AS2Abstract::decode()	57
AS2Abstract::addFile()	57
AS2Abstract::\$message_id	56
AS2Abstract::\$is_signed	55
AS2Abstract::\$filename	55
AS2Abstract::\$adapter	54
AS2Abstract::\$files	55
AS2Abstract::\$headers	55
AS2Abstract::\$is_crypted	55
AS2Abstract::encode()	57
AS2Abstract::generateMessageID()	58
AS2Abstract::getPartnerTo()	59
AS2Abstract::getPartnerFrom()	59
AS2Abstract::getPath()	59
AS2Abstract::getUrl()	59
AS2Abstract::isCrypted()	60
AS2Abstract::getMessageId()	59
AS2Abstract::getHeaders()	59
AS2Abstract::getContent()	58
AS2Abstract::getAuthentication()	58
AS2Abstract::getFileName()	58
AS2Abstract::getFiles()	58
AS2Abstract::getHeader()	58
AS2Adapter::sign()	70
<i>Sign a file which contains mime headers</i>	
AS2Adapter::verify()	70
<i>Verify a signed file</i>	
AS2Header::next()	82
AS2Header::key()	82
AS2Header::offsetExists()	82

AS2Header::offsetGet()	82
AS2Header::offsetUnset()	83
AS2Header::offsetSet()	83
AS2Header::getHeaders()	82

<i>Return all headers as an array</i>	81
AS2Header::getHeader()	81
<i>Return the value of an header</i>	
AS2Header::addHeadersFromMessage()	80
<i>Add a set of headers extracted from a mime message</i>	
AS2Header::addHeaders()	80
<i>Add a set of headers (or override currents)</i>	
AS2Header::count()	81
<i>Return the count of headers</i>	
AS2Header::current()	81

AS2Header::exists()	81
<i>Check if an header exists</i>	
AS2Header::parseHttpRequest()	83
<i>Extract headers from http request and return a new instance of AS2Header</i>	
AS2Header::parseText()	84
<i>Extract headers from mime message and return a new instance of AS2Header</i>	
AS2Log::FAILURE	86
AS2Log::ERROR	86
AS2Log::INFO	86
AS2Log::WARNING	86
AS2Log::\$code	86
AS2Log	86
<i>AS2Secure - PHP Lib for AS2 message encoding / decoding</i>	
AS2Header::toString()	85
<i>Magic method that returns headers serialized as in mime message</i>	
AS2Header::rewind()	84
AS2Header::removeHeader()	84
<i>Remove an header</i>	
AS2Header::setHeaders()	85
<i>Reset all current headers with new values</i>	
AS2Header::toFormattedArray()	85
<i>Return all headers as a formatted array</i>	
AS2Header::valid()	85
AS2Header::addHeader()	79
<i>Add new header (or override current one)</i>	
AS2Header::\$ position	79
AS2Connector	73
<i>AS2Secure - PHP Lib for AS2 message encoding / decoding</i>	
AS2Client::sendRequest()	73
<i>Send request to the partner (manage headers, security, ...)</i>	
AS2Connector::STATUS_ERROR	74
AS2Connector::STATUS_OK	74
AS2Connector::onReceivedMessage()	74
AS2Connector::onReceivedMDN()	74
AS2Client::handleResponseHeader()	73
<i>Allow to retrieve HTTP headers even if there is HTTP redirections</i>	
AS2Client::getLastResponse()	72
<i>Return the last request : headers/content</i>	
AS2Client	71
<i>AS2Secure - PHP Lib for AS2 message encoding / decoding</i>	
AS2Adapter::deleteTempFiles()	71
<i>Garbage collector to delete temp files created with 'self::getTempFilename'</i>	

AS2Client::\$response_content	72
AS2Client::\$response_headers	72
AS2Client::\$response_indice	72
AS2Connector::onSentMDN()	75
AS2Connector::onSentMessage()	75
AS2Exception::getLevel()	78
AS2Exception::\$level_warning	77
AS2Exception::getMessageShort()	78
AS2Header	78
<i>AS2Secure - PHP Lib for AS2 message encoding / decoding</i>	
AS2Header::\$headers	79
AS2Exception::\$level_failure	77
AS2Exception::\$level_error	77
AS2Exception::DEFAULT_ERROR	76
AS2Exception	76
<i>AS2Secure - PHP Lib for AS2 message encoding / decoding</i>	
AS2Exception::STATUS_ERROR	76
<i>Refers to RFC 4130</i>	
AS2Exception::STATUS_FAILURE	76
AS2Exception::STATUS_WARNING	76
AS2Abstract	54
<i>AS2Secure - PHP Lib for AS2 message encoding / decoding</i>	

C

constructor AS2MDN::__construct()	95
constructor AS2Log::__construct()	88
<i>Constructor</i>	
constructor AS2Message::__construct()	98
constructor AS2Partner::__construct()	108
<i>Restricted constructor</i>	
constructor AS2Request::__construct()	111
constructor AS2Header::__construct()	79
constructor AS2Exception::__construct()	77
constructor Horde MIME Part::__construct()	30
<i>MIME_Part constructor.</i>	
constructor AS2Abstract::__construct()	57
constructor AS2Adapter::__construct()	63
constructor AS2Client::__construct()	72
constructor Horde MIME Message::__construct()	21
<i>Constructor - creates a new MIME email message.</i>	

E

ENCOTHER	8
ENCQUOTEDPRINTABLE	8
ENCBINARY	8
ENCBASE64	8
ENC8BIT	8
ENC7BIT	8

H

Horde_MIME_Part::setCharset()	41
<i>Set the character set of this part.</i>	
Horde_MIME_Part::setContentID()	41
<i>Sets the Content-ID header for this part.</i>	
Horde_MIME_Part::setContents()	42
<i>Set the body contents of this part.</i>	
Horde_MIME_Part::setBytes()	41
<i>Explicitly set the size (in bytes) of this part. This value will only be returned (via <code>getBytes()</code>) if there are no contents currently set.</i>	
Horde_MIME_Part::replaceEOL()	40
<i>Replace newlines in this part's contents with those specified by either the given newline sequence or the part's current EOL setting.</i>	
Horde_MIME_Part::getUniqueID()	39
<i>Return the unique MIME_Part boundary string generated for this object.</i>	
Horde_MIME_Part::header()	40
<i>Add the appropriate MIME headers for this part to an existing array.</i>	
Horde_MIME_Part::removePart()	40
<i>Remove a MIME_Part subpart.</i>	
Horde_MIME_Part::setContentTypeParameter()	42
<i>Add a content type parameter to this part.</i>	
Horde_MIME_Part::setDescription()	43
<i>Set the description of this part.</i>	
Horde_MIME_Part::setName()	45
<i>Set the name of this part.</i>	
Horde_MIME_Part::setTransferEncoding()	45
<i>Set the transfer encoding to use for this part.</i>	
Horde_MIME_Part::setType()	45
<i>Set the mimetype of this part.</i>	
Horde_MIME_Part::setMIMEId()	44
<i>Alter the MIME ID of this part.</i>	
Horde_MIME_Part::setInformation()	44
<i>Add information about the MIME_Part.</i>	
Horde_MIME_Part::setDisposition()	43
<i>Set the content-disposition of this part.</i>	
Horde_MIME_Part::setDispositionParameter()	43
<i>Add a disposition parameter to this part.</i>	
Horde_MIME_Part::setEOL()	44
<i>Sets a new string to use for EOLs.</i>	
Horde_MIME_Part::getType()	39
<i>Get the full MIME Content-Type of this part.</i>	
Horde_MIME_Part::getTransferEncoding()	39
<i>Get the transfer encoding for the part based on the user requested transfer encoding and the current contents of the part.</i>	
Horde_MIME_Part::getDefaultType()	35
<i>If the subtype of a MIME part is unrecognized by an application, the default type should be used instead (See RFC 2046). This method returns the default subtype for a particular primary MIME Type.</i>	
Horde_MIME_Part::getDescription()	35
<i>Get the description of this part.</i>	
Horde_MIME_Part::getDisposition()	36
<i>Get the content-disposition of this part.</i>	

Horde MIME Part::getCurrentEncoding()	35
<i>Retrieves the current encoding of the contents in the object.</i>	
Horde MIME Part::getContentParameters()	34
<i>Get a content type parameter from this part.</i>	
Horde MIME Part::getCIDList()	34
<i>Returns the list of CIDs for this part.</i>	
Horde MIME Part::getContentID()	34
<i>Returns the Content-ID for this part.</i>	
Horde MIME Part::getContents()	34
<i>Return the body of the part.</i>	
Horde MIME Part::getDispositionParameter()	36
<i>Get a disposition parameter from this part.</i>	
Horde MIME Part::getEOL()	36
<i>Get the string to use for EOLs.</i>	
Horde MIME Part::getPrimaryType()	38
<i>Get the primary type of this part.</i>	
Horde MIME Part::getRelativeMIMEId()	38
<i>Returns the relative MIME ID of this part.</i>	
Horde MIME Part::getSubType()	39
<i>Get the subtype of this part.</i>	
Horde MIME Part::getParts()	38
<i>Get a list of all MIME subparts.</i>	
Horde MIME Part::getPart()	37
<i>Retrieve a specific MIME part.</i>	
Horde MIME Part::getInformation()	36
<i>Retrieve information about the MIME_Part.</i>	
Horde MIME Part::getMIMEId()	37
<i>Returns the MIME ID of this part.</i>	
Horde MIME Part::getName()	37
<i>Get the name of this part.</i>	
Horde MIME Part::splitContents()	46
<i>Split the contents of the current Part into its respective subparts, if it is multipart MIME encoding. Unlike the <code>imap_*()</code> public functions, this will preserve all MIME header information.</i>	
Horde MIME Part::strict7bit()	46
<i>Should we make sure the message is encoded via 7-bit (e.g. to adhere to mail delivery standards such as RFC 2821)?</i>	
Horde String::length()	117
<i>Returns the character (not byte) length of a string.</i>	
Horde String::lower()	118
<i>Makes a string lowercase.</i>	
Horde String::pad()	118
<i>Returns a string padded to a certain length with another string.</i>	
Horde String::isUpper()	117
<i>Returns true if every character in the parameter is an uppercase letter in the current locale.</i>	
Horde String::isLower()	117
<i>Returns true if ever character in the parameter is a lowercase letter in the current locale.</i>	
Horde String::convertCharset()	115
<i>Converts a string from one charset to another.</i>	
Horde String::extensionExists()	116
<i>Caches the result of <code>extension_loaded()</code> calls.</i>	

Horde_String::isAlpha()	116
<i>Returns true if the every character in the parameter is an alphabetic character.</i>	
Horde_String::pos()	119
<i>Returns the numeric position of the first occurrence of \$needle in the \$haystack string.</i>	
Horde_String::regexMatch()	119
<i>Performs a multibyte safe regex match search on the text provided.</i>	
Horde_String::wrap()	122
<i>Wraps the text of a message.</i>	
Horde_String::convertCharset()	122
<i>Internal public function used to do charset conversion.</i>	
Horde_String::mbstringCharset()	123
<i>Workaround charsets that don't work with mbstring public functions.</i>	
Horde_String::wordwrap()	121
<i>Wraps the text of a message.</i>	
Horde_String::upper()	121
<i>Makes a string uppercase.</i>	
Horde_String::setDefaultCharset()	120
<i>Sets a default charset that the Horde_String:: methods will use if none is explicitly specified.</i>	
Horde_String::substr()	120
<i>Returns part of a string.</i>	
Horde_String::ucfirst()	120
<i>Returns a string with the first letter capitalized if it is alphabetic.</i>	
Horde_String::\$charset	115
Horde_String	115
<i>The Horde_String:: class provides static methods for charset and locale safe string manipulation.</i>	
Horde_MIME_Part::transferEncodeContents()	48
<i>Transfer encode the contents (to the transfer encoding identified via getTransferEncoding()) and set as the part's new contents.</i>	
Horde_MIME_Part::generateBoundary()	48
<i>Generate the unique boundary string (if not already done).</i>	
Horde_MIME_Part::generateIdMap()	48
<i>Generates a mapping of MIME_Parts with their MIME IDs.</i>	
Horde_MIME_Part::transferEncode()	48
<i>Encodes the contents with the part's transfer encoding.</i>	
Horde_MIME_Part::transferDecodeContents()	47
<i>Transfer decode the contents and set them as the new contents.</i>	
Horde_MIME_Part::toCanonicalString()	46
<i>Returns the encoded part in strict RFC 822 & 2045 output - namely, all newlines end with the canonical <CR><LF> sequence.</i>	
Horde_MIME_Part::toString()	47
<i>Return the entire part in MIME format. Includes headers on request.</i>	
Horde_MIME_Part::transferDecode()	47
<i>Decodes the contents of the part to either a 7bit or 8bit encoding.</i>	
Horde_MIME_Part::partFind()	49
<i>Function used to find a specific MIME Part by ID.</i>	
Horde_MIME_Structure	49
<i>\$Horde: framework/MIME/MIME/Structure.php,v 1.87.10.31 2009/01/06 15:23:20 jan Exp \$</i>	
Horde_MIME_Structure::getParameters()	51

	<i>Get all parameters for a given portion of a message.</i>	
Horde MIME Structure:: parse()	<i>Given the results of <code>imap_fetchstructure()</code>, parse the structure of the message, figuring out correct bodypart numbers, etc.</i>	52
Horde MIME Structure:: setInfo()	<i>Given a mime part from <code>imap_fetchstructure()</code>, munge it into a useful form and make sure that any parameters which are missing are given default values.</i>	52
Horde MIME Structure:: convertMimeDecodeData()	<i>Convert the output from <code>mimeDecode::decode()</code> into a structure that matches <code>imap_fetchstructure()</code> output.</i>	51
Horde MIME Structure:: parseTextMIMEMessage()	<i>Attempts to build a <code>MIME_Message</code> object from a text message.</i>	51
Horde MIME Structure:: addMultipartInfo()	<i>Set the special information for certain MIME types.</i>	50
Horde MIME Structure:: parse()	<i>Given the results of <code>imap_fetchstructure()</code>, parse the structure of the message, figuring out correct bodypart numbers, etc.</i>	50
Horde MIME Structure:: parseMIMEHeaders()	<i>Builds an array consisting of MIME header/value pairs.</i>	50
Horde MIME Part:: getCharset()	<i>Get the character set to use for of this part. Returns a charset for all types (not just 'text/*') since we use this charset to determine how to encode text in MIME headers.</i>	34
Horde MIME Part:: getCanonicalContents()	<i>Returns the contents in strict RFC 822 & 2045 output - namely, all newlines end with the canonical <CR><LF> sequence.</i>	33
Horde MIME:: is8bit()	<i>Determines if a string contains 8-bit (non US-ASCII) characters.</i>	16
Horde MIME:: parseAddressList()	<i>Parses a list of email addresses into its parts.</i>	16
Horde MIME:: quotedPrintableEncode()	<i>Encodes a line via quoted-printable encoding.</i>	17
Horde MIME:: generateMessageID()	<i>Generates a Message-ID string conforming to RFC 2822 [3.6.4] and the standards outlined in 'draft-ietf-usefor-message-id-01.txt'.</i>	15
Horde MIME:: encoding()	<i>Returns the MIME encoding for the given input.</i>	15
Horde MIME:: encode()	<i>Encodes a string containing non-ASCII characters according to RFC 2047.</i>	14
Horde MIME:: encodeAddress()	<i>Encodes a string containing email addresses according to RFC 2047.</i>	14
Horde MIME:: encodeRFC2231()	<i>Encodes a string pursuant to RFC 2231.</i>	15
Horde MIME:: rfc822Explode()	<i>Explodes an RFC 2822 string, ignoring a delimiter if preceded by a "\" character, or if the delimiter is inside single or double quotes.</i>	17
Horde MIME:: rfc822WriteAddress()	<i>Builds an RFC 822 compliant email address.</i>	17
Horde MIME:: rfc822Encode()	<i>Quotes and escapes the given string if necessary using rules contained in RFC 2822 [3.2.5].</i>	20

Horde MIME Message	20
<i>The MIME_Message:: class provides methods for creating and manipulating MIME email messages.</i>	
Horde MIME Message::\$ build	21
<i>Has the message been parsed via buildMessage()?</i>	
Horde MIME:: getCode()	19
<i>Retrieves MIME encoding/type data from the internal arrays.</i>	
Horde MIME:: encode()	19
<i>Internal recursive public function to RFC 2047 encode a string.</i>	
Horde MIME::trimEmailAddress()	18
<i>If an email address has no personal information, get rid of any angle brackets (<>) around it.</i>	
Horde MIME::type()	18
<i>Returns the MIME type for the given input.</i>	
Horde MIME::wrapHeaders()	18
<i>Adds proper linebreaks to a header string.</i>	
Horde MIME::decodeRFC2231()	13
<i>Decodes an RFC 2231-encoded string.</i>	
Horde MIME::decodeAddrString()	13
<i>Decodes an RFC 2047-encoded address string.</i>	
Horde Util::cloneObject()	4
<i>Returns an object's clone.</i>	
Horde Util::deleteAtShutdown()	4
<i>Removes given elements at request shutdown.</i>	
Horde Util::extensionExists()	5
<i>Caches the result of extension_loaded() calls.</i>	
Horde Util::\$securedel	4
Horde Util::\$files	4
HORDE_ERROR_DRIVER_CONFIG_MISSING	2
<i>Error code for a missing driver configuration.</i>	
Horde Util	3
<i>The Horde_Util:: class provides generally useful methods of different kinds.</i>	
Horde Util::\$cache	3
Horde Util::getTempFile()	5
<i>Creates a temporary filename for the lifetime of the script, and (optionally) register it to be deleted at request shutdown.</i>	
Horde Util:: deleteAtShutdown()	6
<i>Deletes registered files at request shutdown.</i>	
Horde MIME::addrObject2String()	12
<i>Takes an address object, as returned by imap_header() for example, and formats it as a string.</i>	
Horde MIME::bareAddress()	12
<i>Returns the bare address.</i>	
Horde MIME::decode()	13
<i>Decodes an RFC 2047-encoded string.</i>	
Horde MIME::addrArray2String()	11
<i>Takes an array of address objects, as returned by imap_headerinfo(), for example, and passes each of them through MIME::addrObject2String().</i>	
Horde MIME::\$rfc822 filter	11
<i>Filter for RFC822.</i>	
Horde MIME	10
<i>The MIME:: class provides methods for dealing with MIME standards.</i>	
Horde MIME::\$mime_encodings	10

<i>A listing of the allowed MIME encodings.</i>	
Horde MIME::\$mime_types	11
<i>A listing of the allowed MIME types.</i>	
Horde MIME Message::\$defaultServer	21
<i>The server to default unqualified addresses to.</i>	
Horde MIME Message::buildMessage()	21
<i>Build message from current contents.</i>	
Horde MIME Part::\$subtype	29
<i>The subtype (ex.: plain) of this part.</i>	
Horde MIME Part::\$transferEncoding	29
<i>The desired transfer encoding of this part.</i>	
Horde MIME Part::\$type	30
<i>The type (ex.: text) of this part.</i>	
Horde MIME Part::\$parts	29
<i>The subparts of this part.</i>	
Horde MIME Part::\$mimeid	29
<i>The MIME ID of this part.</i>	
Horde MIME Part::\$flags	28
<i>Internal class flags.</i>	
Horde MIME Part::\$idmap	28
<i>Part -> ID mapping cache.</i>	
Horde MIME Part::\$information	28
<i>Information/Statistics on the subpart.</i>	
Horde MIME Part::addCID()	30
<i>Add to the list of CIDs for this part.</i>	
Horde MIME Part::addPart()	31
<i>Add a MIME subpart.</i>	
Horde MIME Part::getAllContentTypeParameters()	33
<i>Get all parameters from the Content-Type header.</i>	
Horde MIME Part::getAllDispositionParameters()	33
<i>Get all parameters from the Content-Disposition header.</i>	
Horde MIME Part::getBytes()	33
<i>Determine the size of a MIME_Part and its child members.</i>	
Horde MIME Part::contentTypeMap()	32
<i>Returns a mapping of all MIME IDs to their content-types.</i>	
Horde MIME Part::clearContentTypeParameter()	32
<i>Clears a content type parameter from this part.</i>	
Horde MIME Part::alterPart()	31
<i>Alter a current MIME subpart.</i>	
Horde MIME Part::appendContents()	31
<i>Add to the body contents of this part.</i>	
Horde MIME Part::clearContents()	32
<i>Clears the body contents of this part.</i>	
Horde MIME Part::\$eol	27
<i>The sequence to use as EOL for this part.</i>	
Horde MIME Part::\$encode7bit	27
<i>Should the message be encoded via 7-bit?</i>	
Horde MIME Message::header()	23
<i>Add the proper set of MIME headers for this message to an array.</i>	
Horde MIME Message::toString()	24
<i>Return the entire message contents, including headers, as a string.</i>	
Horde MIME Part	24
<i>The MIME_Part:: class provides a wrapper around MIME parts and methods</i>	

	<i>for dealing with them.</i>	
Horde MIME Message::getParts()	Get a list of all MIME subparts.	23
Horde MIME Message::getPart()	Retrieve a specific MIME part.	23
Horde MIME Message::convertMIMEPart()	Create a MIME_Message object from a MIME_Part object.	22
Horde MIME Message::encode()	Take a set of headers and make sure they are encoded properly.	22
Horde MIME Message::getBasePart()	Return the base part of the message. This public function does NOT return a reference to make sure that the whole MIME_Message object isn't accidentally modified.	22
Horde MIME Part::\$ boundary	Unique MIME_Part boundary string.	24
Horde MIME Part::\$ bytes	Default value for this Part's size.	25
Horde MIME Part::\$ description	The description of this part.	26
Horde MIME Part::\$ disposition	The disposition of this part (inline or attachment).	27
Horde MIME Part::\$ dispositionParameters	The disposition parameters of this part.	27
Horde MIME Part::\$ currentEncoding	The current transfer encoding of the contents of this part.	26
Horde MIME Part::\$ contentTypeParameters	The content type parameters of this part.	26
Horde MIME Part::\$ cids	The list of CIDs for this part.	25
Horde MIME Part::\$ contentid	The content-ID for this part.	25
Horde MIME Part::\$ contents	The body of the part.	26
HORDE_ERROR_DRIVER_CONFIG	Error code for an incomplete driver configuration.	2

M

MIME_DEFAULT_ENCODING	The default MIME encoding.	9
MIME_PART_EOL	The character(s) used internally for EOLs.	9
MIME_PART_RFC_EOL	The character string designated by RFCs 822/2045 to designate EOLs in MIME messages.	9
MIME_DEFAULT_DISPOSITION	The default MIME disposition.	9
MIME_DEFAULT_DESCRIPTION	The default MIME description.	9
MIME_CODE	Return a code for type()/encoding().	8
MIME_STRING		8

<i>Return a string for type()/encoding().</i>	
MIME_DEFAULT_CHARSET	9
<i>The default MIME character set.</i>	
MIME.php	8

P

Part.php	9
------------------------------------	---

T

TYPEOTHER	8
TYPETEXT	8
TYPEVIDEO	8
TYPEMULTIPART	8
TYPEMODEL	8
<i>Older versions of PHP's imap extension don't define TYPEMODEL.</i>	
TYPEAUDIO	8
TYPEIMAGE	8
TYPEMESSAGE	8
TYPEAPPLICATION	8

U

Util.php	2
------------------------------------	---