API DLL COSSMTPDLL

The COSSMTPDLL DLL allows send emails from Cosmos.

This utility is available as of Cosmos 7.6.

BASE100

BASE 100, S.A.
www.base100.com

Index

1. Introduction

The COSSMTPDLL DLL allows send emails from Cosmos.

E-mails can be sent in both text and HTML format, with the possibility of attaching files in both formats and, in the case of HTML, also include images in the body of the message.

2. Architecture

The steps to send an email from Cosmos using the COSSMTPDLL library are as follows:

- Creating an email object with the CosSmtpNewMail function.
- Assignment of properties to the email object: SMTP server, subject, sender, receiver, body.
- Sending the email with the CosSmtpSendMail function.
- Freeing the email object with the CosSmtpFreeMail function.

3. Functions

3.1 CosSmtpNewMail

This function creates an email object.

Syntax:

```
CosSmtpNewMail() return integer
```

Parameters:

No parameters.

Returns:

Identifier of the email object. This identifier is the onet that will be used in the other functions of the API to refer to this object.

3.2 CosSmtpFreeMail

This function frees an email object previously create with the CosSmtpNewmail function.

Syntax:

```
CosSmtpFreeMail(mailId as integer) return integer
```

Parameters:

mailId	Identifier of the email object to be released. This identifier is the one returned by the CosSmtpNewMail function.

Returns:

0	The email object has been successfully freed.
-1	The email object has not been released correctly, since the email identifier passed as a parameter does not exist or has been previously released.

3.3 CosSmtpSetUrl

This function assigns the email object the URL and optionally the port of the SMTP server that you want to use to send the email.

Syntax:

```
CosSmtpSetUrl(mailId as integer, url as char) return integer
```

Parameters:

mailId	Identifier of the email object to which you want to assign the property. This identifier is the one returned by the CosSmtpNewMail function.
url	Url of the SMTP server that you want to use to send the email. Optionally, you can indicate the connection port.

Examples:

```
smtp://servidorsmtp.com
smtp://servidorsmtp.com:25
smtps://servidorsmtpseguro.com
smtps://servidorsmtpseguro.com:465
```

Returns:

0 The SMTP server has been successfully assigned to the email object.

-1 The SMTP server has not been correctly assigned to the email object, since the email identifier passed as a parameter does not exist or has been previously freed.

3.4 CosSmtpSetFrom

This function tells the email object the email address to use to send the message.

Syntax:

```
CosSmtpSetFrom(mailId as integer, fromStr as char) return integer
```

Parameters:

mailId Identifier of the email object to which you want to assign the property. This identifier is the one returned by the CosSmtpNewMail function.

fromStr Email address to be used to send the message.

Returns:

0 The email address for sending the message has been assigned correctly.

-1 The email address for sending the message has not been assigned correctly, since the email identifier passed as a parameter does not exist or has been previously released.

3.5 CosSmtpSetSubject

This function tells the email object the subject of the email.

Syntax:

```
CosSmtpSetSubject(mailId as integer, subjectStr as char) return integer
```

Parameters:

mailId Identifier of the email object to which you want to assign the property. This identifier is the one returned by the CosSmtpNewMail function.

subjectStr Message Subject.

Returns:

0 The email subject has been assigned successfully.

-1 The email subject has not been assigned correctly, as the email identifier passed as a parameter does not exist or has been previously released.

3.6 CosSmtpAddRecipient

This function adds the email address of the recipient to the email object. If you want to send the email to more than one recipient, you will have to execute the function once for each recipient.

Syntax:

```
CosSmtpAddRecipient(mailIse d as integer, recipientStr as char, type as inte-
ger) return integer
```

Parameters:

mailId Identifier of the email object to which you want to assign the recipient. This identifier is the one returned by the CosSmtpNewMail function.

recipientStr Character string with the recipient's email address.

type Integer indicating the type of recipient.

Possible values:

(0) Indicates the main recipient or recipients of the email (To).

(1) Indicates the secondary recipient (s) of the email (CC).

(2) Indicates the secondary recipient (s) of the email (blind carbon copy , BCC).

Returns:

0 The recipient of the email has been assigned successfully.

-1 The recipient of the email has not been assigned correctly, as the email identifier passed as a parameter does not exist or has been previously released.

3.7 CosSmtpSetBody

This function assigns the body text of the email.

Syntax:

```
CosSmtpSetBody(mailId as integer, bodyStr as char, fromFile as boolean,
isHTML as boolean, charset as char) return integer
```

Parameters:

mailId Identifier of the email object to which you want to assign the recipient. This identifier is the one returned by the CosSmtpNewMail function.

bodyStr Text of the body of the message.

fromFile Boolean value that will specify whether the body of the message indicated in the bodyStr parameter refers to the absolute path of a text file with the body content (TRUE value), or if, on the contrary, the body text will be the one indicated by the bodyStr parameter (value FALSE).

isHTML Boolean value that will specify if the body of the message is encoded in HTML (and therefore it will be sent in HTML —TRUE value—), or if, on the contrary, the body

	of the message is plain text (and therefore it will be sent as plain text —value FALSE—).
charset	Character string that indicates the character set (charset) in which the body of the message is encoded (“UTF-8”, “ISO8859-1”, etc.).

Returns:

0	The body of the email has been assigned successfully.
-1	The email body has not been assigned correctly, as the email identifier passed as a parameter does not exist or has been previously freed.

3.8 CosSmtpSetAuthUser

This function indicates the user with which the DLL will "login" to the SMTP server indicated in the CosSmtpSetUrl function.

Syntax:

```
CosSmtpSetAuthUser(mailId as integer , user as char) return integer
```

Parameters:

mailId	Identifier of the email object to which you want to assign the "login" user. This identifier is the one returned by the CosSmtpNewMail function.
user	User identifier of "login".

Returns:

0	The "login" user has been assigned correctly.
-1	The "login" user has not been assigned correctly, since the email identifier passed as a parameter does not exist or has been previously released.

3.9 CosSmtpSetAuthPasswd

This function indicates the password of the user specified in the CosSmtpSetAuthUser function.

Syntax:

```
CosSmtpSetAuthPasswd (mailId as integer , password as char) return integer
```

Parameters:

mailId	Identifier of the email object to which you want to assign the "login" user. This identifier is the one returned by the CosSmtpNewMail function.
password	Password of the "login" user.

Returns:

0	The password to the "login" user has been assigned correctly.
-1	The password to the "login" user has not been assigned correctly, since the email identifier passed as a parameter does not exist or has been previously released.

3.10 CosSmtpSetTimeout

This function allows you to establish a connection “timeout” to the SMTP server.

Syntax:

```
CosSmtpSetTimeout(mailId as integer , timeout as integer) return integer
```

Parameters:

mailId	Identifier of the email object to which you want to assign the "timeout". This identifier is the one returned by the CosSmtpNewMail function.
timeout	Timeout in seconds.

Returns:

0	The property has been successfully assigned.
-1	The property has not been assigned correctly, since the email identifier passed as a parameter does not exist or has been previously freed.

3.11 CosSmtpUseSTARTTLS

This function tells the DLL that communication with the SMTP server will be done securely using STARTTLS, which is an extension of the plaintext protocols that allows a plaintext connection to be converted to an en-crypted connection. In order to use this type of secure communication, the SMTP server must implement it and have it active.

The STARTTLS extension is defined for the SMTP protocol in the RFC3207 standard.

Syntax:

```
CosSmtpUseSTARTTLS(mailId as integer , useSTARTTLS as boolean) return integer
```

Parameters:

mailId	Identifier of the email object to which you want to assign the property. This identifier is the one returned by the CosSmtpNewMail function.
useSTARTTLS	Boolean value. If its value is TRUE, it will use the STARTTLS encryption extension in its communication with the SMTP server.

Returns:

0	The property has been successfully assigned.
-1	The property has not been correctly assigned, since the email identifier passed as a parameter does not exist or has been previously freed.

3.12 CosSmtpUseSSL

This function indicates that, if possible, communication with the SMTP server should be over a secure channel (SSL / TLS). If the SMTP server uses the SMTPS protocol, it is necessary to execute this function and the second parameter must be TRUE.

Syntax:

```
CosSmtpUseSSL(mailId as integer , useSSL as boolean) return integer
```

Parameters:

mailId Identifier of the email object to which you want to assign the property. This identifier is the one returned by the CosSmtpNewMail function.

useSSL Boolean value. If its value is TRUE, it will use SSL / TLS encryption in its communication with the SMTP server.

Returns:

0 The property has been successfully assigned.

-1 The property has not been correctly assigned, since the email identifier passed as a parameter does not exist or has been previously freed.

3.13 CosSmtpAddResource

This function allows you to insert resources (images) to an email in HTML format.

To do this, in the HTML code of the email to be sent, the standard HTML tag "cid" with the format «src="cid:resource_name"» must be used.

Syntax:

```
CosSmtpAddResource(mailId as integer, file as char, mimeType as char,
mimeName as char, mimeEncoder as char) return integer
```

Parameters:

mailId Identifier of the email object to which you want to attach the resource. This identifier is the one returned by the CosSmtpNewMail function.

file Full path of the multimedia file to be inserted.

mimeType MIME type of the resource (for example: "image / gif", "image / jpeg", "image / png").

mimeName Resource identifier within the HTML body of the email. This value must match the one indicated in the "cid" label.

mimeEncoder Format in which the resource will be encoded in the email. Its possible values are: "binary", "8bit", "7bit", "base64" and "quoted-printable".

Returns:

0 The property has been successfully assigned.

-1 The property has not been correctly assigned, since the email identifier passed as a parameter does not exist or has been previously freed.

Example: HTML code section of the email where the "part1" resource is found:

```
<td colspan="3"><img
src="cid:part1"
```

```
width="430" height="6"></td>
</tr>
```

Image association to the resource:

```
CosSmtpAddResource(mailId, "c:\mail\logo_firma.jpg", "image/jpeg", "part1",
"base64");
```

3.14 CosSmtpAttachFile

This function allows you to attach a file to the email. This function must be executed once for each file to be attached.

Syntax:

```
CosSmtpAttachFile(mailId as integer, file as char, mimeEncoder as char) return integer
```

Parameters:

mailId Identifier of the email object to which you want to attach the file. This identifier is the one returned by the CosSmtpNewMail function.

file Absolute path of the file to be attached.

mimeEncoder Character string that indicates the encoding method to be used when sending the attached file. If no value is specified, it will be encoded in base 64. Possible values are: “binary”, “8bit”, “7bit”, “base64” and “quoted-printable”.

Returns:

0 The property has been successfully assigned.

-1 The property has not been correctly assigned, since the email identifier passed as a parameter does not exist or has been previously freed.

3.15 CosSmtpSendMail

This function sends the email to the indicated email address.

Syntax:

```
CosSmtpSendMail(mailId as integer) return integer
```

Parameters:

mailId Identifier of the email object. This identifier is the one returned by the CosSmtpNewMail function.

Returns:

0 The email has been sent successfully.

-1 The email identifier passed as a parameter does not exist or has been previously freed.

-2 No sending server specified with the CosSmtpSetUrl function.

-3	No email sender address has been specified with the CosSmtpSetFrom function.
-4	No email receiving address has been specified with the CosSmtpAddRecipient function.
Other value	See table in Annex.

3.16 CosSmtpGetReturnCode

This function returns the SMTP status code of the previously performed email send operation. This code is returned by the SMTP server (not by the DLL), based on some of the RFC standards associated with SMTP communication (RFC821, RFC3463, RFC5248, RFC5321, etc.).

Syntax:

```
CosSmtpGetReturnCode (mailId as integer) return integer
```

Parameters:

mailId	Identifier of the email object. Este identificador es el devuelto por la función CosSmtpNewMail.

Devuelve:

Return code from SMTP operation.

3.17 CosSmtpGetErrorStr

This function returns the text associated with the error code returned by the CosSmtpSendMail function.

Syntax:

```
CosSmtpGetErrorStr(mailId as integer) return char
```

Parameters:

mailId	Email object identifier. This identifier is the one returned by the CosSmtpNewMail function.

Returns:

Text associated with the error code returned by the CosSmtpSendMail function. See list of error codes in Annex.

4. Annex. Error Codes

Code	Description
0	No error.
-1	
-2	Smtp URL can't be NULL. Please, use CosSmtpSetUrl function to set SMTP server.
-3	The sender can't be NULL. Please, use CosSmtpSetFrom function to set mail sender.
-4	The list of recipients can't be NULL. Please, use CosSmtpAddRecipient to set mail recipients.
1	Unsupported protocol.
2	Failed initialization.
3	URL using bad/illegal format or missing URL.
4	A requested feature, protocol or option was not found built-in this libcurl due to a build-time decision.
5	Couldn't resolve proxy name.
6	Couldn't resolve host name.
7	Couldn't connect to server.
8	FTP: weird server reply.
9	Access denied to remote resource.
10	FTP: The server failed to connect to data port.
11	FTP: unknown PASS reply".
12	Time out.
13	FTP: unknown PASV reply.
14	FTP: unknown 227 response format.
15	FTP: can't figure out the host in the PASV response.
16	Error in the HTTP2 framing layer.
17	FTP: couldn't set file type.
18	Transferred a partial file.
19	FTP: couldn't retrieve (RETR failed) the specified file.
20	Obsolete code.
21	Quote command returned error.
22	HTTP response code said error.
23	Failed writing received data to disk/application.
24	Obsolete code.
25	Upload failed (at start/before it took off).
26	Failed to open/read local data from file/application.
27	Out of memory.
28	Timeout was reached.
29	Obsolete code.
30	FTP: command PORT failed.
31	FTP: command REST failed.
32	Obsolete code.
33	Requested range was not delivered by the server.
34	Internal problem setting up the POST.

Code	Description
35	SSL connect error.
36	Couldn't resume download.
37	Couldn't read a file.
38	LDAP: cannot bind.
39	LDAP: search failed.
40	Obsolete code.
41	A required function in the library was not found.
42	Operation was aborted by an application callback.
43	A libcurl function was given a bad argumen.
44	Obsolete code.
45	Failed binding local connection end.
46	Obsolete code.
47	Number of redirects hit maximum amount.
48	An unknown option was passed in to libcurl.
49	Malformed telnet option.
50	Obsolete code.
51	SSL peer certificate or SSH remote key was not OK.
52	Server returned nothing (no headers, no data).
53	SSL crypto engine not found.
54	Cannot set SSL crypto engine as default.
55	Failed sending data to the peer.
56	Failure when receiving data from the peer.
57	Obsolete code.
58	Problem with the local SSL certificate.
59	Couldn't use specified SSL cipher.
60	Peer certificate cannot be authenticated with given CA certificates.
61	Unrecognized or bad HTTP Content or Transfer-Encoding.
62	Invalid LDAP URL.
63	Maximum file size exceded.
64	Requested SSL level failed.
65	Send failed since rewinding of the data stream failed.
66	Failed to initialise SSL crypto engine.
67	Login denied.
68	TFTP: File Not Found.
69	TFTP: Access Violation.
70	Disk full or allocation exceded.
71	TFTP: Illegal operation.
72	TFTP: Unknown transfer ID.
73	Remote file already exists.
74	TFTP: No such user.
75	Conversion failed.

Code	Description
76	Caller must register CURLOPT_CONV_ callback options.
77	Problem with the SSL CA cert (path? access rights?).
78	Remote file not found.
79	Error in the SSH layer.
80	Failed to shut down the SSL connection.
81	Socket not ready for send/recv.
82	Failed to load CRL file (path? access rights?, format?).
83	Issuer check against peer certificate failed.
84	FTP: The server did not accept the PRET command.
85	RTSP CSeq mismatch or invalid CSeq.
86	RTSP session error.
87	Unable to parse FTP file list.
88	Chunk callback failed.
89	The max connection limit is reached.
90	SSL public key does not match pinned public key.