

# COBEMAIL

**Version 3.10**

**User Manual**

## INDICE

Introduction .....	3
Instalation .....	4
Parameters .....	7
How generate the email file.....	8
COBEMAIL.WSS Records .....	9
EMAIL-FROM.....	9
EMAIL-TO.....	10
EMAIL-CC.....	11
EMAIL-BCC .....	12
EMAIL-SUBJECT .....	13
EMAIL-MESSAGE .....	14
EMAIL-ATTACHMENT.....	15
EMAIL-TYPE-TEXT.....	16
EMAIL-TYPE-HTML .....	17
EMAIL-CONFIRM .....	18
EMAIL-TEMPLATE .....	19
EMAIL-VAR.....	20
MEMOS .....	21
MEMO VARIABLES .....	22
EMAIL-BLOCK.....	23
COBEMAIL License.....	24

## Introduction

COBEMAIL, it's a software that can help us in the Email send process since our COBOL programs.

Only we need a little configuration for indicate to COBEMAIL how will be send this emails. Once made this configuration, we need create a small text file with our COBOL program for send our email.

COBEMAIL can send emails in Text format, and HTML format. The COBEMAIL technology is based in HTML Templates.

A Template is a HTML file, can be designed with any Web editor like Front Page, but we can use any other html.

Too, using the HTML format, we can include images embeded in our email, and this images will be send in automatic way, because COBEMAIL parse and analyze our template, and all files needed are included in the email.

The file format for our email es very close to HTML code, and can be seemed like the COBVIEW tags.

For send an email, then need know the special tags, for build our Email File.

If we can build a text file since COBOL, then we can send emails using COBEMAIL. It's all necessary, not need know complicated API, or know difficult protocols, all is using COBOL code.

## Instalation

The installation is very simple, we only need execute the setup installation, and follow the screen instructions.

The follow files will be installed in our installation folder:

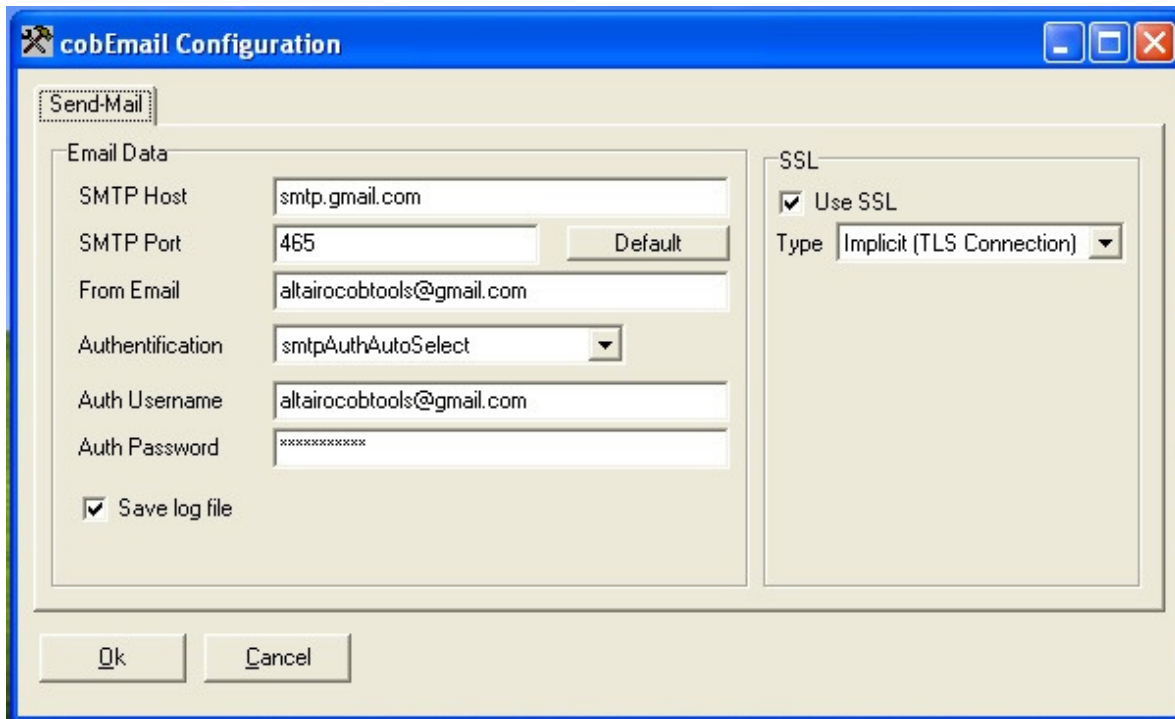
cfgCobEmail.EXE	Configuration program for COBEMAIL
CobSendM.EXE	Program that we execute for send our email
CobEmail.DAT	Configuration file

For all samples in this manual, and the samples that can download from internet, we will assume that our installation director is: C:\COBEMAIL>

Before start to use COBEMAIL, we need configure this, for send emails.

Execute the **cfgCobEmail**, wich configure the COBEMAIL program.

When execute this program, the next screen is showed:



The data that we need are the following:

**SMTP Host** – This data can be a domain name or IP address, and is the SMTP server that we use for send emails. If we do not have a SMTP and an account, we can't send emails.

**SMTP Port** – Is the TCP port where our SMTP server listen the connections, by default we can use the **smtp** port, but several ISP have blocked this port, so we can use the port indicated by our ISP.

**From Email** – Here can indicate the email account since our emails will be sended. The FROM account can be specified in our COBOL file generated.

**Authentication** – Allows specify any Auth method, if our SMTP server need auth our email before to send.

COBEMAIL have the next Auth methods:

- smtpAuthNone
- smtpAuthPlain
- smtpAuthLogin
- smtpAuthCramMD5
- smtpAuthCramSha1
- smtpAuthNtlm
- smtpAuthAutoSelect

The option depends of our SMTP server, and how is configured.

If you server need authentication, and you not know wich, select smtpAuthAutoSelect for test connection.

**Auth Username** – Is the username for Auth in our SMTP server.

**Auth Password** – The password for Username in Auth mode. Since ver. 3.10, this password is saved encrypted in the configuration file.

**Save Log File** – With this data, we can save a log file, for test if our email is sended OK, or have any communications error with the SMTP server.

Is for test only, remember, this log file can be very big if we active this data in production mode.

**SSL** – With this checkbox, we can indicate that SMTP Server is need use SSL security for send email. This depend on how the SMTP server is configured.

Very useful if we go to use GMAIL accounts for send emails.

**SSL Type** – With this opción we can indicate the Security SSL type that our server use.

- None
- Implicit (TLS Connection)
- Explicit (STARTTLS)

The most used is: Implicit (TLS Connection), tha gmail.com servers, although all smtp parameters, this depend on how is configured the SMTP Server.

## Parameters

The program that send our emails is: COBSENDM.EXE  
And the format for execute this program is:

COBSENDM email-file options

Options:

### **/NODEL**

This parameter allos us for indicate that the email-file not be deleted after COBEMAIL send the email.

By default, the email-file is erased after sended.

If we need this file for work with this after send our email, we need indicate /NODEL parameter.

COBEMAIL only delete the email-file, any other file included in the email like ATTACH is not erased, wither our template file, and images needed by our template.

Samples:

1. COBSENDM EMAILTXT
2. COBSENDM.EXE EMAIL.TXT /NODEL
3. COBSENDM.EXE C:\TEMP\EMAIL.TXT

In the sample 1, the email is sended with the indicated instrucciones in the file EMAIL.TXT, this file will be erased after email sended.

With the sample 2, the email is sended, and this file not be erased.

And, in the sample 3, we can indicate the full path for our email file.

We can create our email file in any other directory, and send the full path for this file.

### **/LOG filename.log**

With this parameter we can indicate in which log file save the log records.

By default the logs are saved on cobemail.log in the same path where cobsendm.exe is allocated.

## How generate the email file

For generate our email file, we need declare a file text in our COBOL program.

In the next COBOL code, we declare a text file, like a report file.

```
SELECT EMAILDATA
      ASSIGN TO PRINT "EMAIL.TXT"
      FILE STATUS IS STAT-EMAIL.
```

With this declaration we build a report file with the name: EMAIL.TXT

And for the FD, we can declare like next:

```
FD  EMAILDATA
    LABEL RECORDS ARE OMITTED.
01  R-EMAIL          PIC X(500).
```

Now, with COBEMAIL we have a library for build our email file.

This library is in the COBEMAIL.WSS file

This file have several records for build automatically the tags for our email.

Only declare a COPY "COBEMAIL.WSS" in the WORKING-STORAGE SECTION, for use this records.



## COBEMAIL.WSS Records

### EMAIL-FROM

This record allow us for indicate name and account where we sended the email.

This tag too replace the FROM account, that is indicated in the configuration global.

This tag is optional, because if we not indicate EMAIL-FROM record, COBEMAIL use the FROM account from the configuration.

For use this record, we need send 2 values, in the next variables:

FROM-NAME – Indicate the person name, or account description.

FROM-EMAIL – Is the account for send the email.

Sample:

```
MOVE "ALBERTO ORTEGA" TO FROM-NAME.  
MOVE "aortega@cobtools.com" TO FROM-EMAIL.  
WRITE R-EMAIL FROM EMAIL-FROM.
```

With this COBOL instrucciones, in our email file, generate the next tag:

```
<FROM NAME =' ALBERTO ORTEGA' EMAIL=' aortega@cobtools.com' >
```

## EMAIL-TO

With this record we can specify for wich account is sendes the email.

We need indicate the name, and email account.

The next data is necessary for use this record.

TO-NAME – Name or description for email account to send email.

TO-EMAIL – Is the email account for the email receiver.

Sample:

```
MOVE "JUAN PEREZ" TO TO-NAME.  
MOVE "jperrez@dominio.com" TO TO-EMAIL.  
WRITE R-EMAIL FROM EMAIL-TO.
```

With this COBOL instrucciones, we generate the next TAG in our email file:

```
<TO NAME =' JUAN PEREZ' EMAIL=' jperrez@dominio.com' >
```

We can use this record several times, if we need send several copies for the same email.

## EMAIL-CC

This record is used for indicate copies for our email.

The data need are:

CC-NAME – Name or account description for send a copy from the email.

CC-EMAIL – Email account for receive a email copy.

Sample:

```
MOVE "MIGUEL HERNANDEZ" TO CC-NAME.  
MOVE "mhernandez@dominio.com" TO CC-EMAIL.  
WRITE R-EMAIL FROM EMAIL-CC.
```

TAG generated in our email file:

```
<CC NAME ='MIGUEL HERNANDEZ' EMAIL='mhernandez@dominio.com' >
```

## EMAIL-BCC

This record is used for indicate Blind Carbon Copy, or hidden accounts for our email.

The data need are:

BCC-NAME – Name or account description for send a copy from the email.

BCC-EMAIL – Email account for receive a email copy.

Sample:

```
MOVE "MIGUEL HERNAN" TO BCC-NAME.  
MOVE "mhernan@dominio.com" TO BCC-EMAIL.  
WRITE R-EMAIL FROM EMAIL-BCC.
```

TAG generated in our email file:

```
<BCC NAME ='MIGUEL HERNAN' EMAIL='mhernan@dominio.com' >
```

## EMAIL-SUBJECT

This record is used for send the SUBJECT for the email, is only a record needed.  
If we use several subject records, the last will be used like subject.

For use this record, only need move the value to SUBJECT variable

Sample:

```
MOVE "REPORT SALES FOR 01/2006" TO SUBJECT.  
WRITE R-EMAIL FROM EMAIL-SUBJECT.
```

In our email file, we can have the next TAG:

```
<SUBJECT CONTENT='REPORT SALES FOR 01/2006' >
```

## EMAIL-MESSAGE

This record is used for write our email body to send in the email.

We can use this record several times, for write several body lines.

This record is only for TEXT email.

For use this record, only move the line to MESSAGE variable

Sample:

```
MOVE "LINE IN THE EMAIL BODY" TO MESSAGE.  
WRITE R-EMAIL FROM EMAIL-MESSAGE.  
  
MOVE "ANOTHER EMAIL LINE" TO MESSAGE.  
WRITE R-EMAIL FROM EMAIL-MESSAGE.
```

In our email file, generate the next TAGS:

```
<MESSAGE CONTENT='LINE IN THE EMAIL BODY'>  
<MESSAGE CONTENT=ANOTHER EMAIL LINE'>
```

## EMAIL-ATTACHMENT

This record is a very useful advantage for COBEMAIL, with this record we can indicate a filename to send like attach in our email.

We can indicate several files in the same email.

Only move the filename to FILE-ATTACH variable.

We can indicate the fullpath, o relative path, always remember in wich directory are we allocated before the call COBSENDM.EXE program.

Sample:

```
MOVE "C:\TEMP\DATA.ZIP" TO FILE-ATTACH.  
WRITE R-EMAIL FROM EMAIL-ATTACHMENT.  
  
MOVE "F:\DATOS\SEND.DAT" TO FILE-ATTACH.  
WRITE R-EMAIL FROM EMAIL-ATTACHMENT.
```

The TAGS generated with this code is:

```
<ATTACHMENT FILE='C:\TEMP\DATA.ZIP' >  
<ATTACHMENT FILE='F:\DATOS\SEND.DAT' >
```

## EMAIL-TYPE-TEXT

With this record we can indicate that, our email will be sended in TEXT format.

Is not variable necessary

Sample:

```
WRITE R-EMAIL FROM EMAIL-TYPE-TEXT.
```

TAG generated in our email file:

```
<EMAIL TYPE='TEXT' >
```

Whe you send a email in TEXT format, the body for this email is only in simpel text.

This body text can be indicated with the record EMAIL-MESSAGE, y not be used the record EMAIL-TEMPLATE for indicate a HTML email template.



## EMAIL-TYPE-HTML

This record is for indicate that our email will be sended in HTML format

Is not variable necessary

Sample:

```
WRITE R-EMAIL FROM EMAIL-TYPE-HTML.
```

TAG generated in our email file:

```
<EMAIL TYPE=' HTML' >
```

When an email in HTML format is sended, the body for the email in the EMAIL-MESSAGE record can be in text or html format.

But, the best way for send an email in html format, is using a HTML template, builded with any HTML editor.

## EMAIL-CONFIRM

With this record we can solicit a confirmation that our email is received.

The confirmation is sended to our email account that we used in the record EMAIL-FROM.

If the client email can't support this feature for send responses, we never receive our confirmation.

Sample:

```
WRITE R-EMAIL FROM EMAIL-CONFIRM.
```

## EMAIL-TEMPLATE

Allow us use a HTML template file to use.

This template file is in HTML format, all images embeded in the page too send like attach in the email automatically.

This images that our template include, will be included, searching in the next directory and order:

- 1.- In the fullpath that is indicated in the HTML template
2. - In the same path where is allocated our template file
3. - In the same path where is allocated our COBSENDM.EXE program

Once that our image is allocated, is inserted for send in our email.

Sample:

```
MOVE ".\HTML\EMAIL1.HTM" TO FILE-TEMPLATE.  
WRITE R-EMAIL FROM EMAIL-TEMPLATE.
```

With this COBOL code, the next TAG is inserted:

```
<TEMPLATE FILE='.\HTML\EMAIL1.HTM' >
```

In this sample, we indicated that our template file is alloacted in the directory .\HTML\, with the EMAIL1.HTML name.

## EMAIL-VAR

This record allow us indicate a value for any variable declared inside our Template file.

Several times we need write in our email any data, like Name person, Address, and any other variable parth.

By sample, if we write a letter for many Customers, we can indicate the Customer Name like variable, for put in this variable the Customer name send by our COBOL program.

Sample:

```
MOVE "NAME" TO EMAIL-VAR-NAME .
MOVE WSS-NOMBRE TO EMAIL-VAR-VALUE .
WRITE R-EMAIL FROM EMAIL-VAR .
```

The EMAIL-VAR-VALUE have a X(250) picture, if we need send any numeric value, we need first mask this value with another variable.

In our template file, we can indicate the variable name, in format **<? Variable ?>**

For this sample, we need indicated like: **<? NAME ?>**

We can use several variables, like we need.

If we indicate a variable in our template, but we can't assign value in our program, this variable is showed empty or blank space.

The program TEST3.CBL for the COBEMAIL samples, have a test like how use variables in templates.

## MEMOS

This records are for work with multi-line information.

This MEMOS are for include text in the email body, and can't be used in HTML templates email.

Need work with several record in your COBOL program.

Sample:

```
WRITE LINEA FROM EMAIL-START-MEMO .  
  
MOVE "LINE 1" TO EMAIL-DAT-MEMO .  
WRITE R-EMAIL FROM EMAIL-MEMO .  
  
MOVE "LINE 2" TO EMAIL-DAT-MEMO .  
WRITE R-EMAIL FROM EMAIL-MEMO .  
  
WRITE LINEA FROM EMAIL-END-MEMO .
```

The record EMAIL-START-MEMO allow for indicate the begin for the Memo content.

The record EMAIL-END-MEMO is used for finish the Memo.

The EMAIL-DAT-MEMO variable is for write the MEMO to send in the Email.  
Can include the CRLF chars in the variable.

The EMAIL-DATA field is 1024 characters size.

We can use the sequence `\r\n` for insert Line Break manually.

## MEMO VARIABLES

This records are for work with info in several lines and store this content in any var.

This MEMOS we can use in HTML templates, and work like vars.

In the sample we can see the several record to use.

Sample:

```
MOVE "NOM-VAR" TO EMAIL-VAR-MEMO .
WRITE LINEA FROM EMAIL-START-VAR-MEMO .

MOVE "LINE 1" TO EMAIL-DAT-MEMO .
WRITE R-EMAIL FROM EMAIL-MEMO .

MOVE "LINE 2" TO EMAIL-DAT-MEMO .
WRITE R-EMAIL FROM EMAIL-MEMO .

WRITE LINEA FROM EMAIL-END-VAR-MEMO .
```

First, we need indicate the variable name to use, in the EMAIL-VAR-MEMO field.

The record EMAIL-START-VAR-MEMO allow for indicate the begin for the Memo content.

The record EMAIL-END-VAR-MEMO is used for finish the Memo.

The EMAIL-DAT-MEMO variable is for write the MEMO to send in the Email.  
Can include the CRLF chars in the variable.

The EMAIL-DATA field is 1024 characters size.

We can use the sequence `\r\n` for insert Line Break manually.

This memo can be used in the Template HTML file, like variables, to write the content, we use:

**<? NOM-VAR ?>**

## EMAIL-BLOCK

With this record we can build repeat blocks in our HTML template.  
By sample, when we need send any list of data to our customer, like invoices.

The invoices, is a rows list, where we can repeat N times the same data.  
In any customer have 2,3 invoices, but any other can have 100 invoices.

Sample:

```
MOVE "INVOICE" TO EMAIL-BLOCK-NAME.  
MOVE "FACT_PASTE" TO EMAIL-BLOCK-PASTE.  
WRITE R-EMAIL FROM EMAIL-BLOCK.
```

With this code, we indicate that want write a **BLOCK** called **INVOICE**, that need exist in our template, and we need paste in any place, indicated by our tag **FACT\_PASTE**.

Now, inside our HTML template, the blocks are defined in the following way:

```
<!--? paste fact_paste ?-->  
<!--? block invoice ?-->  
<tr>  
    <td><? date ?></td>  
    <td><? concept ?></td>  
    <td><? import ?></td>  
</tr>  
<!--? endblock invive ?-->
```

Very important, define the names that we use in the EMAIL-BLOCK record, if we use another names in our template, our email not work.

The only way for write this tags is using `<!--? ?-->`, that are comments in HTML mode.

Too is very important that if in our block we write variables, like in our sample, we need indicate the values for this variables before write EMAIL-BLOCK record.

In the TEST4.CBL program, have a BLOCKS sample.

## COBEMAIL License

When you acquire your COBEMAIL licence, this is send you in text file.

This license file is need copy in the same directory where is installed the COBEMAIL program.

For buy your license, visit: [www.cobtools.com](http://www.cobtools.com), where you find more information about prices.