

NAME

check_read_email - Check for emails and store them for processing.

DESCRIPTION

Synopsis:

check_read_email.py

Description:

This program will check for emails at regular intervals and if any found it will read and store the messages. When all emails have been saved including any attachments, it starts the script that processes the saved messages.

This program is intended to be run continuously in background. The correct way to stop this program is by sending it a SIGINT signal. That way the program terminates properly. The interval between checks is configurable in the common configuration file.

Arguments:

None

Return:

0 -- Normal exit

Run info:

This program is to be run in background. It is started after a reboot and started and stopped via CRON entries.

See also:

dotenv -- dotenv file with the secret information in.

check_read_email.sh -- Script that executes this program.

kill_check_email.sh -- Script to terminate this program.

process_messages.sh -- Script to process the saved emails.

signage.conf -- Configuration data providing environment variables.

Version:

1.0

Copyright (C) 2023 Gert Bakker, Coleshill, UK

This file is part of the <Digital Signage> project.

Digital Signage 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, version 3.

Digital Signage 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 this program. If not, see <<https://www.gnu.org/licenses/>>.

FUNCTIONS

check_mail_auth(from_adrs)

Check that sender email is in the authorised list.

Check that the sender email address is in the list of authorised email addresses. The checked email address is between the <..> brackets in the from_adrs of the email message.

Arguments:

from_adrs -- Text of the From line of the email message.

Return:

True if in the list, False if not in the list.

clean(text)

Clean text by replacing non-alphanumeric characters with underscore. This is used to clean up the attachment filenames.

All non alphanumeric characters in the given text are replaced with an underscore. It will also preserve the dot character which is essential for file extensions.

Arguments:

text -- The text to be cleaned

Return:

string -- Cleaned text string.

close_email_link()

Close the email link

Close the IMAP email link and logout from the mailbox.
Set the open flag to False.

Arguments:

None

Return:

False -- Always return with False

delete_emails(rem_list)

Remove the emails as per the given list of email IDs

Delete the emails as per the given list of messages IDs.

Arguments:

rem_list -- List of email IDs to delete from the email inbox

return:

None

initialise()

Initialise the program.

Setup the variables that are also used by other functions as GLOBAL. Extract from the Dotenv file the configuration variables and get the DEBUG flag. Configure the logging function with the desired logging level (depends on the DEBUG flag) and the message format. Set the signal capture for the SIGINT signal.

Arguments:

None

Return:

None

main()

Main control of this program

Perform the initialisation routine to ensure that all constants are obtained from the dotenv file and the logging level set to include or exclude debug logging messages.

The -outer- loop is a perpetual loop that opens the email inbox. Any problems with this will then ensure that the inner loop is not executed and will sleep for the configurable amount of seconds. The latter is to ensure that there is a wait before the connection to the email server is attempted again.

Then the -inner- loop is started. This loop is only run if the 'sentry' flag is true. Therefore this flag controls the loop.

This inner loop is the one that will check if there are any INBOX email messages. If messages found, read and save the messages and any attachments. Then process the email messages.

At the end of this -inner- loop, wait for a "configurable" number

of seconds before continuing the loop.

Arguments:
None

Return:
None

open_email_inbox()

Open the link to the email inbox and logon to the email account.

Setup the link to the mail server and then logon to the INBOX using the username and password from the dotenv configuration file.
Set the variable "imap" as a global variable as it will be used in other functions.
If it is all successful, set the imap_open flag to True.

Arguments:
None

Return:
True -- Connected and logged on successfully
False -- Problem with connecting to the IMAP server.

read_emails(messages)

Read the emails, store in a file and download any attachments.

Get the message count value from the DOTENV file and save the value to trigger the updating of the DOTENV counter at the end of this function.

Read each email, store its contents into a file and download any attachments. Filenames of attachments that have the designation "=?UTF-8?" in the name are translated to proper filenames.
Messages are stored in the "messages" folder and any attachments are stored in the "attachments" folder.

Arguments:
messages - Number of messages to retrieve

Return:
None

signal_term_handler(signalnumber, frame)

Process the SIGINT signal.

This function is called when the SIGINT signal has been sent to this program. It will log a INFO message to indicate that the signal was received. If the IMAP connection was open, close the connection. This makes it neat and tidy.
Terminate this program with exit code 0 as this is a normal termination.

Arguments:
signalnumber -- The number of the signal that was caught
frame -- Info about when the signal was caught. Ignore this.

Return:
Exit the program with code 0.

sleep(...)

sleep(seconds)

Delay execution for a given number of seconds. The argument may be a floating point number for subsecond precision.