

# TbUtil SMTP Plugin

30 Nov 2020

Chris Lowth <[chris.lowth@turbonomic.com](mailto:chris.lowth@turbonomic.com)>

# Why is the SMTP plugin needed?

- XL (multiple deployment models) and classic have different email sending mechanisms.
- In some cases we need to install multiple Linux packages to send email from scripts.
- A standardized way of sending emails from scripts would be helpful.

# What is it?

- A way to send emails from Turbonomic integration scripts.
- A new plugin for "TbUtil".
  - Written in "GO"
  - Can be use in TbScript scripts (written in JavaScript).
- It comes with version 1.2c and later.
- It has no dependencies other than "TbUtil" itself.
- Please consider the 1.2c version to be "experimental".
  - Future releases will productionise it.

# Example use case.

- Email report on status of targets
- Logic:
  - Collect information about targets using `getTargets()` API call.
  - Write to a local HTML file `"/tmp/targets.html"`.
  - Load the SMTP plugin
  - Load the email configuration file (we'll use `"email-config.json"`)
  - Call the plugin's `"send"` function with the required configuration data.

# "send(...)" function configuration

- Takes a JS object argument containing..
  - Optionally: SMTP server config details.
    - Server IP and port, User, password or secret, encryption type, sender email address
    - The config from Turbo itself can optionally be used.
  - Email header fields
    - Subject
    - To, Cc, Bcc
  - Body parts list. Can contain:
    - Embedded text
    - Attachments.
  - Consider building it using the "template(..)" function.

# Example Script

```
var smtp = plugin("smtp-plugin");
```

Load the plugin and get a reference to it.

```
var targets = client.getTargets();
```

Collect the data and format the HTML report.

```
// code here to create HTML file "/tmp/targets.html"  
// would go here (omitted for brevity).
```

```
var info = {  
    date: (new Date()).format("Mon Jan 2 2006"),  
    filePath: "/tmp/targets.html",  
    fileName: "targets.html"  
};
```

Read the config file, and patch in the date and file name.

Parse the file's text (which must be in JSON format) into a JS object.

```
var configText = template("email-config.json", info);  
var configObject = JSON.parse(configText);
```

Send the email.  
*Specifying the API client object as the 2<sup>nd</sup> argument allows the email server details to be taken from the Turbo instance itself.*

```
smtp.send(configObject, client)
```

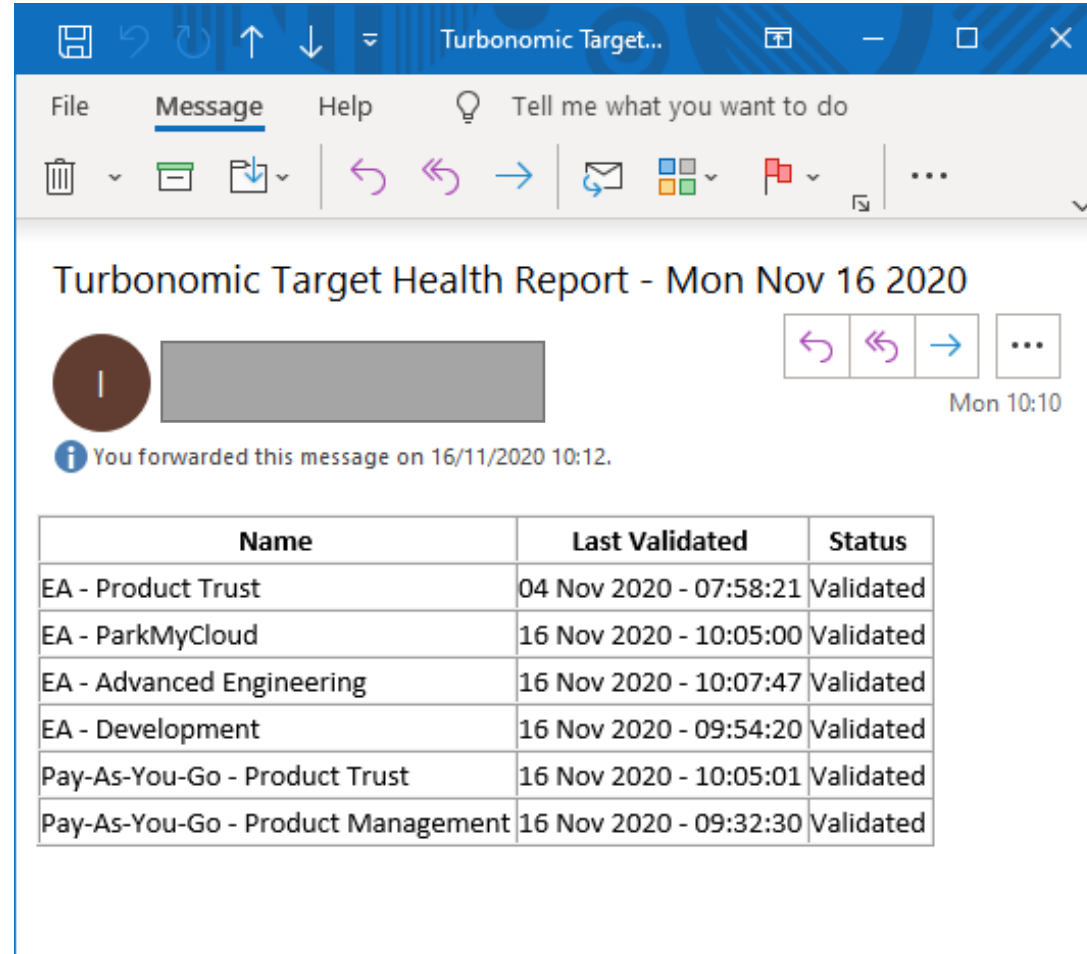
# Example config file for HTML message

- This uses config from Turbo itself
- It sends the html text in the body of the message to two people.







```
{  
  "to": "chris@example-domain.com",  
  "cc": "mark@example-domain.com",  
  "subject": "Target Health Report - {{ .date }}",  
  "parts": [  
    {  
      "mimeType": "text/html",  
      "encoding": "quoted-printable",  
      "contentFile": "{{ .filePath }}",  
    }  
  ]  
}
```

Expanded by the  
"template()" function  
in the JS code


# Received HTML email



Turbonomic Target Health Report - Mon Nov 16 2020

Mon 10:10

 You forwarded this message on 16/11/2020 10:12.

Name	Last Validated	Status
EA - Product Trust	04 Nov 2020 - 07:58:21	Validated
EA - ParkMyCloud	16 Nov 2020 - 10:05:00	Validated
EA - Advanced Engineering	16 Nov 2020 - 10:07:47	Validated
EA - Development	16 Nov 2020 - 09:54:20	Validated
Pay-As-You-Go - Product Trust	16 Nov 2020 - 10:05:01	Validated
Pay-As-You-Go - Product Management	16 Nov 2020 - 09:32:30	Validated



# Example config for message with attachment

- uses config from Turbo itself
- sends the html text as attachment to the body of the message

```
{  
  "to": "chris@example-domain.com",  
  "subject": "Target Health Report - {{ .date }}",  
  "parts": [  
    {  
      "mimeType": "text/plain",  
      "content": "Here's the report you wanted"  
    },  
    {  
      "mimeType": "text/html",  
      "encoding": "base64",  
      "contentFile": "{{ .filePath }}",  
      "fileName": "{{ .fileName }}"  
    }  
  ]  
}
```



Expanded by the "template()" function in the JS code

The diagram shows a yellow box with the text "Expanded by the 'template()' function in the JS code". Three blue arrows originate from this box and point to the template placeholders in the JSON: one to the subject's date field, one to the filePath field, and one to the fileName field.

# Documentation

- See the smtp-plugin.pdf file, available on the TbUtil download page and in the installation tar file.