EMsoft:
Setting Up User Notifications

Release 3.2, August 23, 2017

Table of Contents

1 User Notifications in EMsoft

Some of the programs in the EMsoft package can take quite a while to run. In version 3.2, we have added an option to have those programs automatically notify the user when the run has completed. The notification can be either in the form of an email message, or via a message push to a slack.com channel. Access to this new feature only requires a few additional lines in the EMsoftConfig.json configuration file.

1.1 Email notifications

Email notification is very easy to set up and only requires one additional configuration parameter:

"EMNotify": "Email",

In addition, you must make sure that the UserEmail parameter has the correct email address to which the notifications should be sent.

1.2 Slack notifications

Slack notifications require that the user set up a slack.com message channel. To do so, first obtain a team Slack URL of the form your-team-url.slack.com; even if you are just working by yourself, you will still need to set up a team. You could, for instance, name your team in the following form: institution-EMsoft.slack.com, where institution is the abbreviation for your university or research entity. Then you provide your email address and a password to complete the set up of your slack account (this is the free account).

Once you have a team URL, you will find that you have two default message channels, #general and #random; we suggest that you create an #EMsoft channel as well. At this point, you may want to also install the desktop and/or mobile slack app which you can download from the slack.com site or via the Apple Store for Mac OS X users. You can access your slack.com account via a web browser or via the app. You will also need to allow slack to push notifications to your desktop/mobile.

Next, you will need to set up incoming webhook integration. Point your browser to

https://api.slack.com/custom-integrations

Select the Incoming Webhooks menu item. In the first section titled *Send data into Slack in real-time*, click on the incoming webhook integration link in the paragraph starting with *Start by setting up* .... You will get a page titled Incoming Webhooks; select the name of the channel that you wish to use for messaging from the pull down menu in the *Post to Channel* form. Then click the green button, which will get you to an integration settings page. From this page you will need to copy the Webhook URL entry into you EMsoftConfig.json configuration file; this is a long entry with several code strings in it that uniquely define your message channel, so you should not share this URL with anyone else (unless you want that person to be able to post messages to your channel).

In the EMsoftConfig.json file, add the following three lines:

"EMNotify": "Slack",
"EMSlackWebHookURL": "https://hooks.slack.com/services/T6.../B6...",
"EMSlackChannel": "channelname",

where channelname must be replaced by the name of the channel you want messages to be posted to (without the # character). On the *Integration Settings* page, check the other items and customize as needed, then Save the Settings. You should now have a functioning EMsoft message channel! Try it by running the EMsoftSlackTest program from the command line...

2 Adding notifications to your own f90 programs

If you are developing EMsoft code, you can easily add notification support to your code. Here is a simple example of code that you could insert into your own subroutine or main program:

```
! at the top of your subroutine
use notifications

! in the variable declaration section
character(fnlen),ALLOCATABLE      :: MessageLines(:)
integer(kind=irg)                 :: NumLines
character(fnlen)                  :: MessageTitle
character(100)                    :: c

! at the end of your program
if (trim(EMsoft_getNotify()).ne.'Off') then
  if (trim(namelist%Notify).eq.'On') then
! allocate the message string array
    NumLines = 2
    allocate(MessageLines(NumLines))
! get the hostname
    call hostnm(c)
! initialize the message stings
    MessageLines(1) = 'ProgramName program has ended successfully'
    MessageLines(2) = 'Data stored in '//trim(outname)
! define the title of the message (will appear in bold face at the top of your message)
    MessageTitle = 'EMsoft on '//trim(c)
! and post the message to Slack
    i = PostMessage(MessageLines, NumLines, MessageTitle)
  end if
end if
```

In this code the variable "outname" is a full path file name. The variable *namelist%Notify* is part of a namelist that was read from a namelist parameter file; see the EMMCOpenCL.template file for an example. Obviously, you can add more output lines by increasing the *NumLines* parameter and entering appropriate text in the *MessageLines* string array. You could print out the values of certain program variables, for instance the execution time, or the final results of the calculation.

When executed from a workstation with IP name mycomputer.cmu.edu, this code will produce a Slack message of the following form:

```
EMsoft on MYCOMPUTER.CMU.EDU APP [1:50 PM]
ProgramName program has ended successfully
Data stored in /Users/.../Files/.../testfile.h5
```

Finally, we should note that a free slack.com account entitles you to a maximum of $10,000$ messages; you can manually delete messages one by one, but to remove that restriction you will have to register with a paid account. Slack makes it nearly impossible to perform bulk deletes of messages by limiting how many individual delete commands can be requested per second.