If the log.html file is automatically generated in the workspace root after running the Robot Framework script, you don't need to explicitly provide its path. You can adjust the script to account for this and only specify the path for manual-alerts.xlsx. Here's the updated pipeline script:

**Updated Pipeline Script**

pipeline {

agent any

stages {

stage('Run Robot Scripts') {

steps {

// Run Robot Framework scripts

echo "Running Robot Framework scripts..."

}

}

}

post {

success {

script {

def attachmentPaths = 'robot/TM/generic-modules/xl-lib/manual-alerts.xlsx,log.html'

emailext(

subject: "Build #${env.BUILD\_NUMBER} - Success!",

body: """<p>The build was successful.</p>

<p>Please find the attached files for details:</p>

<ul>

<li>Manual Alerts Report (manual-alerts.xlsx)</li>

<li>Execution Log (log.html)</li>

</ul>""",

to: 'recipient@example.com',

attachmentsPattern: attachmentPaths,

mimeType: 'text/html'

)

}

}

failure {

emailext(

subject: "Build #${env.BUILD\_NUMBER} - FAILED!",

body: """<p>The build has failed. Please check the logs.</p>""",

to: 'recipient@example.com',

mimeType: 'text/html'

)

}

}

}

**Key Updates**

1. **Attachment Paths**:
   * Included log.html directly without specifying its path, as it's located in the root of the workspace.
   * Combined with the path for manual-alerts.xlsx as robot/TM/generic-modules/xl-lib/manual-alerts.xlsx,log.html.
2. **File Handling**:
   * Assumes log.html is generated automatically by the Robot Framework and resides at the root of the Jenkins workspace.
3. **Email Content**:
   * Updated the body to reference both files clearly.

**Notes**

* **Dynamic File Validation**: Ensure both files (manual-alerts.xlsx and log.html) exist at the specified locations when the email is sent.
* **Wildcard Alternative**: You could also use a wildcard pattern if there are variations in the file naming:
* attachmentsPattern: 'robot/TM/generic-modules/xl-lib/manual-alerts.xlsx,\*.html'

This will attach any .html file from the workspace.

Let me know if you need any further refinements!

The error in def attachmentPaths could occur if the pipeline syntax isn't correctly handling Groovy variables or if the emailext function isn't properly formatted. In Jenkins' declarative pipeline, you don't need to declare def for a variable in the script block if it’s not required elsewhere.

Here’s the corrected version of your script:

**Corrected Pipeline Script**

pipeline {

agent any

stages {

stage('Run Robot Scripts') {

steps {

// Run Robot Framework scripts

echo "Running Robot Framework scripts..."

}

}

}

post {

success {

script {

emailext(

subject: "Build #${env.BUILD\_NUMBER} - Success!",

body: """<p>The build was successful.</p>

<p>Please find the attached files for details:</p>

<ul>

<li>Manual Alerts Report (manual-alerts.xlsx)</li>

<li>Execution Log (log.html)</li>

</ul>""",

to: 'recipient@example.com',

attachmentsPattern: 'robot/TM/generic-modules/xl-lib/manual-alerts.xlsx,log.html',

mimeType: 'text/html'

)

}

}

failure {

emailext(

subject: "Build #${env.BUILD\_NUMBER} - FAILED!",

body: """<p>The build has failed. Please check the logs.</p>""",

to: 'recipient@example.com',

mimeType: 'text/html'

)

}

}

}

**Fixes Applied**

1. **Removed def attachmentPaths**:
   * Directly passed the attachment pattern string to the emailext function. Groovy in Jenkins pipelines sometimes has issues with explicitly declared variables like def in script blocks unless they’re used elsewhere.
2. **Inline attachmentsPattern**:
   * The attachmentsPattern is directly inlined within the emailext call.
3. **Post-conditions**:
   * Ensures success and failure blocks handle email notifications correctly without variable dependency.

**Testing**

1. Make sure log.html is generated in the workspace root by the Robot Framework.
2. Confirm that manual-alerts.xlsx exists at the path robot/TM/generic-modules/xl-lib/manual-alerts.xlsx.

Let me know if the issue persists!