z/TPF native code coverage sample scripts readme

Contents
_____

This file includes the following information:

1.0 Introduction
_____

With the Clean as You Code approach to code quality, your focus is always on new code,
which is the code that was added and changed from your previous version. You must ensure
that your new code is clean and safe.

To view code coverage results for the new and changed code for your z/TPF project, you
must first run a baseline scan and create the project in SonarQube. SonarQube identifies
new and changed code by comparing the baseline scan and any scans that follow.

The steps in the following sections walk you through creating the baseline, running a code
coverage, and then running a scan to import the code coverage results. A link to SonarQube
is also provided for you to view the new code analysis. You can then identify how much
your new and changed code is covered by tests and any code execution paths that you might
have missed.

2.0 Change history
_____

20210205   Initial version
20210218   Added Git server as a prerequisite
20210329   Update in retrieving the regKey from jsonResponse due to change in ccv start
           service in apar PJ46420.

3.0 Prerequisites
_____

  - Ensure that APARs PJ46334 and PJ46420 are applied on your z/TPF system.
  - Deploy z/TPF scriptable code coverage services on the z/TPF test system.
  - Configure and start the tpfccv service.
  - Install SonarQube with SonarQube C++ plugin (Community). You are not required to
    install it on IBM on Linux on Z.
  - On Linux on Z, install SonarScanner , and configure it to connect to the SonarQube
    server by editing the sonar-scanner.properties file.
  - On Linux on Z, install Python 3.4 or later with the following Python modules:
    argparse, configparser, getpass, json, os, requests, subprocess, sys,
    textwrap, time

_____

- On Linux on Z, install the Git server

4.0 Installation
_____

1) Use FTP to transfer the tpfNativeCodeCoverage.tar.gz to your linux on Z system. You can
   place the file in any directory as a holding location,

   for example: /tmp/ztpftar

2) Create a root directory to hold the extracted files,

   for example: /ztpftools

3) Extract the files from the tpfNativeCodeCoverage.tar.gz file by entering the following
   commands:

   cd /ztpftools
   tar -xvzf /tmp/ztpftar/tpfNativeCodeCoverage.tar.gz

   The extracted files are in the following directory
   structure:

      tpfNativeCodeCoverage
      |
      |-- tpf_run_native_ccv.properties
      |
      |-- tpf_run_native_ccv.py
      |
      `-- tpf_scan_project.py

4) Verify that the read and execute permissions are set for these files.

   For example, enter the following command to set read and execute permissions for all
   users:

   chmod -R 755 tpfNativeCodeCoverage

5.0 Customizing the scripts
_____

Edit the tpf_scan_project.py file to update the value for the sonarScanner variable with
the correct path to where your sonarScanner is installed.

6.0 Running the scripts
_____

1) Open the command line interface, and change the current directory to the TPF project
   directory:

   cd /home/<user_id>/<tpf_project_directory>

2) Copy the tpf_run_native_ccv.properties file in to your TPF project directory from the
   scripts location.

3) Edit the tpf_run_native_ccv.properties file to update the values for the z/TPF test
   system and TPF project specific variables.

4) Enter the following command to run the baseline sonar scan on your project:

   tpf_scan_project.py

5) Check your SonarQube dashboard to verify the project was created by using the URL that
   was displayed after the baseline scan in the previous step.

_____

6) Make code updates to your project source, and build and load your project to your z/TPF
   test system.

7) Enter the following command to start code coverage on your z/TPF test system:

   tpf_run_native_ccv.py start

8) Run your tests on your z/TPF test system.

9) Enter the following command to stop code coverage on your z/TPF test system:

   tpf_run_native_ccv.py stop

10) Enter the following command to get the formatted code coverage result file in
    Cobertura format from your z/TPF test system:

   tpf_run_native_ccv.py get

11) Enter the following command to run the Sonar scan on your project:

   tpf_scan_project.py

12) Check your SonarQube dashboard to verify that the project displays new code stats by
    using the URL that was displayed after the scan in the previous step.

13) Repeat steps 6 to 12 as often as needed.

7.0 Known problems and workarounds
_____

1) Sometimes you might receive a 400 HTTP status with a native code coverage get operation
   if the stop code coverage post processing is still in progress. So, for the get
   operation to be successful, you need to wait for a few seconds after the stop
   operation before you perform the get operation. To prevent the get operation from
   failing, you can adjust the seconds_to_sleep variable in the tpf_run_native_ccv.py
   file.

2) If you need to run a baseline scan again for your project, complete the following
   steps:

   a) Delete the following files from your project directory:
      sonar-project.properties
      .git
      .scannerwork

   b) Delete the project from SonarQube. This action requires the administrator privilege
      for the SonarQube server.

   c) Run the baseline scan again.

8.0 Other sources of information
_____

1) IBM z/TPF Knowledge Center: https://www.ibm.com/support/knowledgecenter/en/SSB23S
2) SonarQube and SonarScanner: https://www.sonarqube.org/
3) SonarQube C++ plugin (Community): https://github.com/SonarOpenCommunity/sonar-cxx
4) Cobertura format: http://cobertura.sourceforge.net/xml/coverage-04.dtd
5) Git server: https://git-scm.com/

9.0 Notices
_____

This information was developed for products and services offered in the US.

_____

9.1 Trademarks

registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.

The registered trademark Linux® is used pursuant to a sublicense from the Linux Foundation the exclusive licensee of Linus Torvalds, owner of the mark on a world¬wide basis.

9.2 Warranty

This package is provided on an "as is" basis. There are no warranties, express or implied, including the implied warranties of merchantability and fitness for a particular purpose. IBM has no obligation to provide service, defect correction, or any maintenance for the package. IBM has no obligation to supply any updates or enhancements for the package to you even if such are or later become available.