

Blackduck and Veracode Integration

Veracode Scan for MSBuild

Reference URL of MSBuild Jenkins groovy file: - https://tlvgit03.nice.com/ActimizeDeployer/cs-solution/-/blob/testing/pipeline/cs_solution_generic_build/Jenkins_veracode_generic.groovy

Used this code of block for Veracode Scan in MSBuild jobs.

1. Used parameters for "Veracode Scan" if this is true then the above steps should get run inside **dir** specified.
1. **appArtifactId = "\${env.JOB_BASE_NAME}"** will get read from job base name and get stored inside appArtifactId.
1. **appVersion = "4.6.1"** for now it's hardcoded as this app version fetching is under discussion and gets stored inside appVersion.
1. **zipFile = "\${appArtifactId}.zip"** is the full name of a zip that starts from job name.
1. Now using Powershell create the zip by the below command and the file extension containing .pdb and .dll will be zipfile.

Compress-Archive -Path .\assemblies//**/**.pdb, .\assemblies/**/**/**.dll "\${env:JOB_BASE_NAME}.zip" -Force**

1. The most important environment variables are the name of the Veracode project name so it should be start from **ACT_FMC_CS_PROJECT_NAME** and below is the example of the environment variable to be passed.

appName = "ACT_FMC_CS_\${appArtifactId}"

1. Now we have every environment set for the runtime then we can execute Veracode scan to the respected projects with the below: -

```
dir("build") {  
    echo "Using zipFile=${zipFile}, appName=${appName}"  
    veraCodeScan(this, "${appName}", "${workspace}/build/${zipFile}", "java", false, "DevSandbox", true, "/home/ec2-user")  
}
```

Note:- We are running the same Veracode configuration which was referred to in SP jobs and we have made some changes for MSBuild Veracode integration.

Veracode Scan For Java

Reference URL of Java Jenkins groovy file: - https://tlvgit03.nice.com/ActimizeDeployer/cs-solution/-/blob/testing/pipeline/cs_solution_generic_build/Jenkins_java_veracode_blackduck.groovy

Used this code of block for Veracode Scan in Java jobs.

1. Used parameters for "Veracode Scan" if this is True then the above steps should get run inside **dir** specified.
1. **appArtifactId = "\${env.JOB_BASE_NAME}"** will get read from job base name and get stored inside appArtifactId.
1. **appVersion = "4.6.1"** for now it's hardcoded as this app version fetching is under discussion and get stored inside appVersion.
1. **zipFile = "\${appArtifactId}.zip"** is the full name of a zip that starts from job name.
1. Now using Powershell create the zip by the below command and create zipfile of the target folder.

Compress-Archive -Path ./maven/**/target "\${env:JOB_BASE_NAME}.zip" -Force

1. The most important environment variables are the name of the Veracode project name so it should start from **ACT_FMC_CS_PROJECT_NAME** and below is the example of the environment variable to be passed.

appName = "ACT_FMC_CS_\${appArtifactId}"

1. Now we have every environment set for the runtime then we can execute the Veracode scan to the respected projects with the below: -

```
dir("build") {  
  
    echo "Using zipFile=${zipFile}, appName=${appName}"  
  
    veraCodeScan(this, "${appName}", "C:/workspace/FMC/CS/${appArtifactId}/build/${zipFile}", "java", false,  
"DevSandbox", true, "/home/ec2-user")  
  
}
```

Note:- We are running the same Veracode configuration which was referred to in SP Java jobs.

Blackduck Scan for Msbuild

Reference URL of Msbuild Jenkins groovy file: - https://tlvgit03.nice.com/ActimizeDeployer/cs-solution/-/blob/testing/pipeline/cs_solution_generic_build/Jenkins_veracode_generic.groovy

Used this code of block for Blackduck Scan in MSBuild jobs.

1. Used parameters for "Blackduck Scan" if this is True then the above steps should get run inside **dir** specified.
1. **venvFolderName = "\${env:JOB_BASE_NAME}"** will get read from job base name and get stored inside appArtifactId.
1. **appVersion = "4.6.1"** for now it's hardcoded as this app version fetching is under discussion and get stored inside appVersion.
1. **zipFile = "\${appArtifactId}.zip"** is the full name of a zip which starts from job name.
1. Now using Powershell creating the zip by below command and creating the zip of all packages installed.

Compress-Archive -Path ./solutions/**/packages/* "\${env:JOB_BASE_NAME}.zip" -Force

1. The most important environment variables are the name of Blackduck project name so it should be start from **ACT_FMC_CS_PROJECT_NAME** and below is the example of environment variable to be passed.

appName = "ACT_FMC_CS_\${appArtifactId}"

1. Now we have every environment set for the runtime then we can execute Blackduck scan to the respected projects with the below: -

```
dir("build") {  
  
    blackduckScan("${appName}", "${appVersion}", "${WORKSPACE}/build/${zipFile}")  
  
}
```

Note:- We are running the same Blackduck configuration which was referred to in SP jobs and we have made some changes for MSBuild Blackduck integration.

Blackduck Scan for Java

Reference URL of Java Jenkins groovy file: - https://tlvgit03.nice.com/ActimizeDeployer/cs-solution/-/blob/testing/pipeline/cs_solution_generic_build/Jenkins_java_veracode_blackduck.groovy

Used this code of block for Blackduck Scan in Java jobs.

1. Used parameters for "Blackduck Scan" if this is True then the above steps should get run inside **dir** specified.

1. **appArtifactId** = "\${env.JOB_BASE_NAME}" will get read from job base name and get stored inside appArtifactId.

1. **appVersion** = "4.6.1" for now it's hardcoded as this app version fetching is under discussion and get stored inside appVersion.

1. **zipFile** = "\${appArtifactId}.zip" is the full name of a zip which starts from job name.

1. Now using Powershell creating the zip by below command and creating the zipfile of target directory.

Compress-Archive -Path ./maven//target "\${env.JOB_BASE_NAME}.zip" -Force**

1. The most important environment variables are the name of Blackduck project name so it should be start from

ACT_FMC_CS_PROJECT_NAME and below is the example of environment variable to be passed.

appName = "ACT_FMC_CS_\${appArtifactId}"

1. Now we have every environment set for the runtime then we can execute Blackduck scan to the respected projects with the below: -

```
dir("build") {  
    blackduckScan("ACT_${appName}", "${appVersion}", "C:/workspace/FMC/CS/${appArtifactId}/build/${zipFile}")  
}
```

Note:- We are running the same Blackduck configuration which was referred to in SP Java jobs.

Important points: -

- For Veracode/Blackduck we need to create a project which should also match the name of Jenkins job like below and also mentioned in the 6th points of every above documentation:-

1. ACT_FMC_CS_NiceTradingRecording_Configuration_Interface

1. ACT_FMC_CS_ApplicationPlugins_Fusion_NtrClustersStatus

1. ACT_FMC_CS_ApplicationPlugins_Fusion_CallSearch

1. ACT_FMC_CS_SystemOverview_Webservice