Fortify SCA Translation

Cookbook – SCA 18.20





Translation of Source Code that does not Require a Compiler 4

List of Non-Compiled Source Code Languages. 4

General SCA Translation for Compiled Code Questions 5

Do I have access to all the Source Code ? 5

Do I have all the build/compile related libraries for the application ? 5

Java Translation Basics 6

SCA Java Language Translation Common Errors 6

SCA Log Files Changes in 18.20 6

General Translation Tips 7

Key Translation Onboarding Questions for Java/JSP Applications 7

SCA Export Translation For Scan on Another System 7

SCA Translation Log file recommendations 7

Recommended Verify your translation. 7

Java Translation Examples 8

Command Line Translation 8

Key Points for Java Translation 8

Example Translation With different Language types. 8

Java Build Tool Examples 9

Maven Copy JAR files translation example 9

Translate Java with Maven with the Forify SCA Plugin 10

Common Error Message Maven Plugin Not Installed 10

Common Maven Errors JARs missing from POM.xml or Maven M2 Repository 11

Example Error Message Snippet 11

Gradle Translation 12

Known SCA Gradle Plugin JAR Reference Issue 12

Workaround for Gradle Copy Jars 12

Example Workaround Commands. 12

Java Jenkins Command Line Translation 13

Common Errors with Java Translation 14

Invalid CLASSPATH Error Messages: 14

Example Incorrect or missing CLASSPATH 14

JSP Pages Java Related Common Errors: 15

JSP releated Classpath Missing JAR Reference 15

Fortify Java Translation using Azure DevOps Online 15

Fortify SCA Visual Studio Plugin on the Microsoft TFS/Azure DevOps Marketplace 16

Translating Java Applications using TFS/Azure DevOps Tasks Plugin 17

SCA Translation Environment Variables 19

CLASSPATH Environment Variable 19

Changing the default SCA Java JRE in use. 19

Linux/Unix/Mac OS Example 19

Example Windows: 19

Visual Studio Translation Basics 20

List of arguments for command line without plugin. 20

Recommended Install SCA VS Plugin on All SCA Translation Build systems. 20

Errors in Translation. 20

SCA will translate successfully for the following versions: 20

Visual Studio Translation Arguments File Example: 21

Appendix 25

Example Verify a Translation Script. 25

Example Simple Java Translation Script 26

NOTE: This document will not duplicate what is written in the Official documentation.

The official documentation URL:

https://www.microfocus.com/documentation/fortify-static-code-analyzer-and-tools/1820/

# Translation of Source Code that does not Require a Compiler

There are a number of programming languages that Fortify can translate with just the source files.

Fortify translate files based on the file extension of the filename.

## List of Non-Compiled Source Code Languages.

* ActionScript - .as, .mxml - The –flex-libraries command line option needs to be set. See SCA User Guide Page 64.
* Coldfusion - .cfm,.cfml, .cfc - The –source-base-dir command line option needs to be set.   
  See SCA User Guide Page 66.
* Javascript - .js
* Typescript - .ts
* PHP - .php
* HTML - .html,.htm
* Ruby - .rb
* SAP ABAP - .abap
* Visual Basic - .vb6
* VB Script - .vbs, vbscript
* XML - .xml, .xsd, wsdl, .plist, .wsdd, .cpx. .xcfg, .cscfg, .wadcfg, .csdef, .appxmanifest.
* SQL Files – NOTE: The default is Oracle PLSQL on non-windows Systems or TSQL on Windows. You can set the fortify sca command line option –sql-language to specify the correct one for your application. See the SCA User Guide Page 67.

# General SCA Translation for Compiled Code Questions

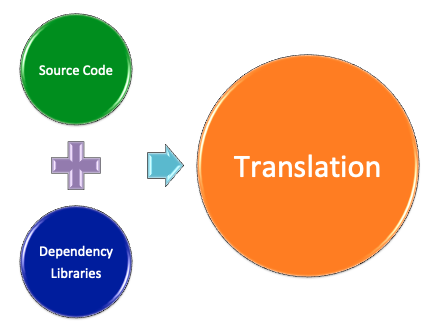
## Do I have access to all the Source Code ?

The most complete translation will access to as much of the applications source code as possible.

## Do I have all the build/compile related libraries for the application ?

The SCA translation process for compiled related code (C/C++, Java, C#/ASP.Net … ), requires that all referenced libraries be locally installed on the system that is performing the translation.

The SCA translation process needs the library files referenced in the build/compile process to be on a local file system path. This may require that you ensure the referenced libraries are copied locally before you can translate without missing references.



# Java Translation Basics

SCA will translate successfully when you have all the Java Source code and do not have to reference any JAR files.

SCA will translate successfully on the command line when you set the CLASSPATH environment variable correctly.

SCA will translate successfully on command line when you set the CLASSPATH with the –cp or –extdirs options correctly

SCA will translate successfully for the following versions:

* Java 5.x
* Java 6x
* Java 7x,
* Java 8x
* Java (including Android) for all of the above versions.

Example:

sourceanalyzer -b myjava8 -source 1.8 -logfile myjava8Translate.log <my java source code directory>

NOTE: The -source option specifies the version of java to translate. This is needed if there are features of a version of java that may have changed from the default 1.8 version. NOTE Java 9 if no longer supported by Oracle.

# SCA Java Language Translation Common Errors

There are six common errors with a translation

1. SCA does not support the Java feature or version
2. SCA does not have read access to the files or directory
3. SCA runs out of system resources
4. SCA translate command line option missing or incorrect
5. SCA translate using build tools has some missing references ( JARs)
6. SCA has a bug for that function or feature.

## SCA Log Files Changes in 18.20

NOTE: See Page 115 in the SCA User Guide for more details.

There are now two log files for each translation starting with version 18.20, a sca.log and a log for fortify support called, sca\_FortifySupport.log.

# General Translation Tips

## Key Translation Onboarding Questions for Java/JSP Applications

The following questions should be asked when setting up a JAVA/JSP application for SCA translation.

1. Where are the Application dependency JAR files located before and during the build process ?
2. Do I have all the source code written by my developers for the application and any third party source that is used in the application that is available ?
3. Do I have all the dependency JAR files locally on my SCA translation system ?
4. How do I ensure that the Java CLASSPATH is set correctly to point to all the required dependency JAR files during the SCA translation process ?

## SCA Export Translation For Scan on Another System

The SCA translation dose not require as much resources as the SCA scan process. Exporting a build translation session to another system for scanning that has better processing power with more memory and better processors is a common best practice.

The command line options –make-mobile and –export-build-session are used to move a translation to another machine for scanning.

The –make-mobile option ensures that all of the source code used in the translation will be packaged in the exported build session. The default mode is to only place the source code related to the issues found into the build session, which limits the ability to analyze issues completely, because only snippets are saved.

Example Translation and Export:

sourceanalyzer –b javaapp1 -clean

sourceanalyzer –b javaapp1 -cp /lib/externalLib.jar:/ lib2/more.jar /src/javaapp1

sourceanalyzer –b javaapp1 -make-mobile

sourceanalyzer –b javaapp1 –export-build-session javaapp1.mbs

## SCA Translation Log file recommendations

Always use the –debug option and the –clobber-log option to capture any issues with the translation that you may need to resolve or send to Fortify Support. You should have a process to archive log files for each scan by datetime.

NOTE: See Page 115,116 in the SCA User Guide for more details.

**Example**:

sourceanalyzer -debug -clobber-log -b javaapp1 -logfile myTrans.log < my source code directory>

## Recommended Verify your translation.

After a translation has completed you should run the –show-files option and verify that all the files you expected to translate were translated. A file will NOT be scanned for issues, if it does NOT appear in the listing from the -show-files option.

NOTE: In the Appendix is a Shell Script example of how you could get a list of files that were not translated from the source directory. This is a demo example only.

# Java Translation Examples

## Command Line Translation

NOTE: For more info See pages 24-30 in the user guide for more details.

Below is a translation for an application found in the source directory /src/javaapp1 with dependency JAR files of /lib/externalLib.jar and /lib2/more.jar.

**Unix/Mac OS/Linux**

sourceanalyzer –b javaapp1 -clean

sourceanalyzer –b javaapp1 -cp /lib/externalLib.jar:/ lib2/more.jar /src/javaapp1

sourceanalyzer –b javaapp1 -make-mobile

sourceanalyzer –b javaapp1 –export-build-session javaapp1.mbs

**Windows**

sourceanalyzer –b javaapp1 -clean

sourceanalyzer –b javaapp1 -cp C:\lib\externalLib.jar;C:\ lib2\more.jar c:\src\javaapp1

sourceanalyzer –b javaapp1 -make-mobile

sourceanalyzer –b javaapp1 –export-build-session javaapp1.mbs

NOTE: The java applications requires a source code list or directory and any referenced JAR files specified in the environment variable CLASSPATH or on the command line with the –cp option.

NOTE: You can use the –extdirs option to point to directories with JAR files in them.

## Key Points for Java Translation

The –clean process should be done before the start of a new translation to ensure that no leftover source code that may have been removed in the development process is still in the SCA translation area.

The translation command after the clean has been performed can be called more than once if needed with the same build ID to add translation from another language type that may be part of the complete application.

## Example Translation With different Language types.

sourceanalyzer –b javaapp1 -clean

sourceanalyzer –b javaapp1 -cp /lib/externalLib.jar:/ lib2/more.jar /src/javaapp1

sourceanalyzer –b javaapp1 /src/htmlAndJavascriptSource

sourceanalyzer –b javaapp1 gcc Cprogram.c

sourceanalyzer –b javaapp1 -make-mobile

sourceanalyzer –b javaapp1 –export-build-session javaapp1.mbs

NOTE: The above commands perform translation on Java code and then appends two other language types, the translation of html/javascript code and a C program compiled with gcc.

# Java Build Tool Examples

SCA can translate source code using the following build tools. There are limitations to each of the build tool support and the following should be understood.

The build tool must be able to generate a complete CLASSPATH to the required JAR files for the javac compiler locally before a SCA translation can successful.

NOTE: There are ways to use build tools in a way that works for the build process, but does not generate a CLASSPATH that the SCA translation process requires for all the source code that it is given to translate.

## Maven Copy JAR files translation example

**Example**:

mvn clean

mvn dependency:copy-dependencies -DoutputDirectory=/tmpjars

sourceanalyzer -b mavenDemoWithJars -clean

sourceanalyzer -b mavenDemoWithJars -extdirs /tmpjars srcdir

**Discussion**

The solution above will use the maven copy dependencies option to download all the referenced jar file in the POM.xml to one directory and then set the JAVA CLASSPATH to that directory using the –extdirs option.

NOTE: This is a workaround for some project that may not work with the Maven Plugin. The key goal is to get all the JARs local and set the CLASSPATH for SCA translation.

This approach works when all the JARs versions referenced are at the same version in all the child pom.xml files. There are cases where some child pom.xml use an older version of the same open source library in some of the pom.xml files.

## Translate Java with Maven with the Forify SCA Plugin

NOTE: First install the Maven Plugin on all the build systems using Maven.

See Page 75 in the SCA user guide for details on installing the Maven plugin.

**Example**:

The following commands runs the maven command to install all of the required JAR files into the Maven local .m2 repository and then translates the source code using the maven POM.xml files and the SCA Maven plugin.

mvn clean install

sourceanalyzer -b mvndemo -clean

sourceanalyzer -b mvndemo mvn clean package

or

mvn clean install

mvn com.fortify.sca.plugins.maven:sca-maven-plugin:18.20:clean

mvn com.fortify.sca.plugins.maven:sca-maven-plugin:18.20:translate

**Discussion**

You must run the mvn clean install before using the plugin to translate. This will ensure that all the JAR reference files are on the local filesystem in the .m2 directory.

The Fortify SCA plugin does not support network http file paths in the CLASSPATH for translation.

The Fortify Maven Plugin using the maven copy classpath option to build the CLASSPATH for the SCA translation. Maven will fail if the dependency JAR files are not installed in the local Maven .m2 repository. Maven supports network http JAR file paths, but the Fortify SCA plugin does NOT.

## Common Error Message Maven Plugin Not Installed

If you see the following error message then you do not have the SCA Plugin installed correctly.

**Example**:

sourceanalyzer -b mvndemo mvn clean package

[INFO] Scanning for projects...

Downloading from central: https://repo.maven.apache.org/maven2/com/fortify/sca/plugins/maven/sca-maven-plugin/maven-metadata.xml

[INFO] ------------------------------------------------------------------------

[INFO] BUILD FAILURE

[INFO] ------------------------------------------------------------------------

[INFO] Total time: 1.503 s

[INFO] Finished at: 2018-11-15T18:04:17-06:00

[INFO] ------------------------------------------------------------------------

[ERROR] Error resolving version for plugin 'com.fortify.sca.plugins.maven:sca-maven-plugin' from the repositories [local (/Users/zacharylewis/.m2/repository), central (https://repo.maven.apache.org/maven2)]: Plugin not found in any plugin repository -> [Help 1]

[ERROR]

[ERROR] To see the full stack trace of the errors, re-run Maven with the -e switch.

[ERROR] Re-run Maven using the -X switch to enable full debug logging.

[ERROR]

[ERROR] For more information about the errors and possible solutions, please read the following articles:

[ERROR] [Help 1] http://cwiki.apache.org/confluence/display/MAVEN/PluginVersionResolutionException

## Common Maven Errors JARs missing from POM.xml or Maven M2 Repository

The Fortify SCA Maven plugin assumes that all reference JAR files are in the Maven local .m2 directory. All Maven referenced projects should be installed locally in the .m2 maven repository to resolve issues like the following.

This error may occur if the build system does not have all the dependency JAR files installed in the same locations as the system the original working build was copied from.

### Example Error Message Snippet

…[ERROR] Failed to execute goal on project mavendemo: Could not resolve dependencies for project com….

## Gradle Translation

See Page 75 in the Users Guide for information on Gradle support.

NOTE: Gradle has many tasks and even custom tasks that Fortify SCA may not support.

The supported Gradle tasks for Java is the JAVA related tasks.

### Known SCA Gradle Plugin JAR Reference Issue

The following error is shown with the Gradle plugin because of a bug that should have a patch soon.

sourceanalyzer -b g2 gradle clean build

…

TaskListener registered.

> Task :clean

[error]: Unable to resolve symbol 'Security' at (/Users/zacharylewis/gitdirs/javatips/examples/chp3/gradle1/src/main/java/demo/program.java:10:25)

> Task :compileJava

### Workaround for Gradle Copy Jars

A workaround is to add a tasks called ScaCopyDependencies to copy all dependency JARs to a tmp directory and then call sourceanalyzer with the –cp or –extdirs CLASSPATH setting to the tmp directory JARs.

Example Tasks.

task ScaCopyDependencies(type: Copy) {

from configurations.default

into 'scadependencies'

}

### Example Workaround Commands.

The following workaround required the addeding of a tasks to copy all JAR dependency files to a directory call scadependencies and then set the classpath to point to the JARs in the scadependencies directory before running the SCA translation process.

gradle ScaCopyDependencies

export CLASSPATH=scadependencies/myrefslib.jar

sourceanalyzer –b gradleworkaround -clean

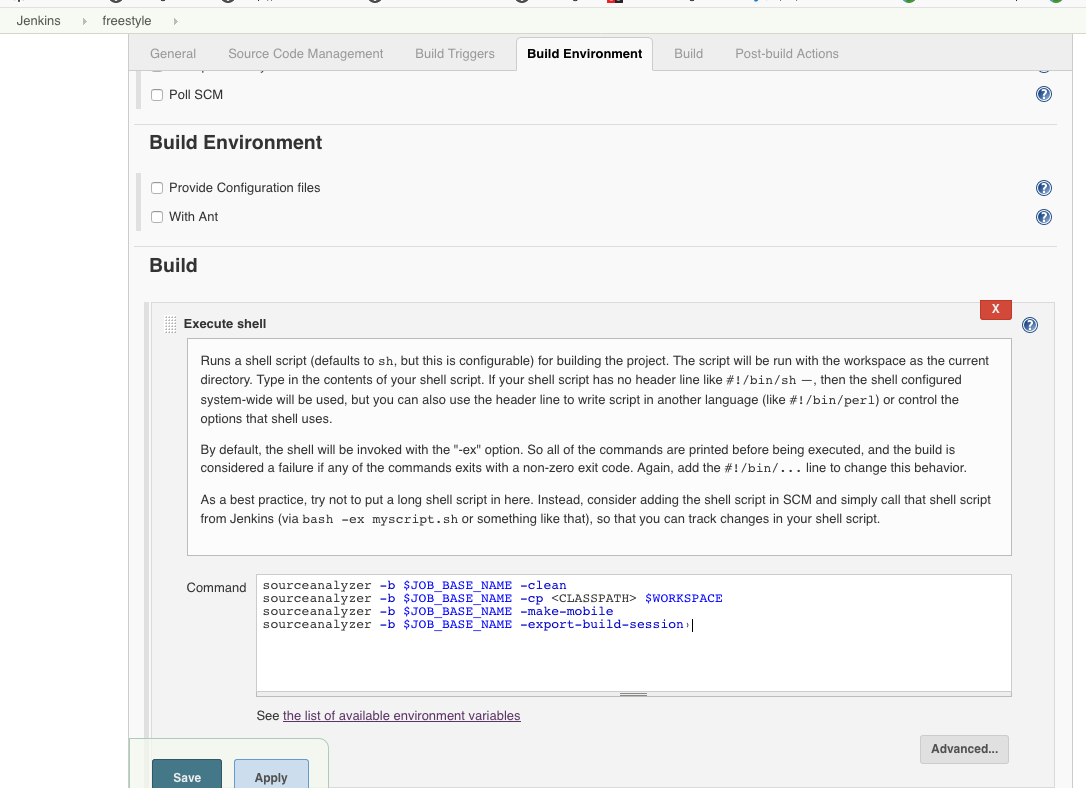
sourceanalyzer –b gradleworkaround -extdirs scadependencies src

sourceanalyzer –b gradleworkaround -make-mobile

sourceanalyzer –b gradleworkaround –export-build-session gradleworkaround.mbs

# Java Jenkins Command Line Translation

Jenkins SCA translation can be with a command line task or using the Jenkins SCA Plugin. The recommended method is to use the command line task and to create a shell or batch script that can be reused by the applications.



# Common Errors with Java Translation

The translation of JAVA source code requires that a CLASSPATH be given for all external dependency JAR files. If this is missing you will see errors similar to the following

.

## Invalid CLASSPATH Error Messages:

Example: (program.java:8:31)

[warning]: The following references to Java functions could not be resolved. These functions may be part of classes that could not be found, or there may be a type error at the call site of the given function relative to the function declaration. Please ensure the Java source code can be compiled by a Java compiler.

getExternalFunction

Java Translation with Dependency JAR files.

Java source code that reference dependency JAR files must specify a CLASSPATH to all the referenced JAR files to translate completely. If the CLASSPATH is incorrect or missing then you will see error messages like the following:

### Example Incorrect or missing CLASSPATH

Unable to locate a class for import org.springframework.web.context.request.async.CallableProcessingInterceptorAdapter

logger:com.fortify.frontend.translator.java.JavaResolver marker:USER thread:sourceanalyzer-13

MDC:{class=org.springframework.samples.mvc.async.TimeoutCallableProcessingInterceptor, frontend=JavaFrontEnd, msgId=1216, prefix=[warning]: , severity=WARNING, sourceInfo=TimeoutCallableProcessingInterceptor.java:8:15:1, stderr=true, step=SRC\_PARSE} NDC:[]

[2018-11-13 14:07:42.431 WARN 1216]

# JSP Pages Java Related Common Errors:

JSP Pages are converted to Java code when the translation process is performed.

## JSP releated Classpath Missing JAR Reference

[2018-11-12 09:47:56.346 WARN 12003]

Assuming Java source level to be 1.8 as it was not specified. Note that the default value may change in future versions.

logger:com.fortify.sca.frontend.JavaFrontEnd marker:USER thread:sourceanalyzer-13

MDC:{frontend=JavaFrontEnd, msgId=12003, severity=WARNING, step=SRC\_PARSE} NDC:[]

[2018-11-12 09:47:56.831 WARN 12022]

The class "javax.servlet.http.HttpServlet" could not be found on the classpath, but it was found in the JAR file provided by Fortify in "C:\Program Files\Fortify\Fortify\_SCA\_and\_Apps\_18.20\Core\default\_jars\javax.servlet-api-3.0.1.jar" as a convenience. To ensure consistent translation behavior add the JAR file that contains "javax.servlet.http.HttpServlet" to the classpath given to the translation step. Refer to the documentation about "default JARs" in the SCA User Guide for more information.

logger:com.fortify.messaging.MessageManager marker:USER thread:sourceanalyzer-13

MDC:{class=JSPPAGE.\_.\_jspXSS\_jsp, frontend=JavaFrontEnd, msgId=12022, pass=Resolving, severity=WARNING, sourceInfo=XSS.jsp:1:1, step=SRC\_PARSE, webapp=C:\zacwork\nodejsdemo\demoXss\web} NDC:[]

## Fortify Java Translation using Azure DevOps Online

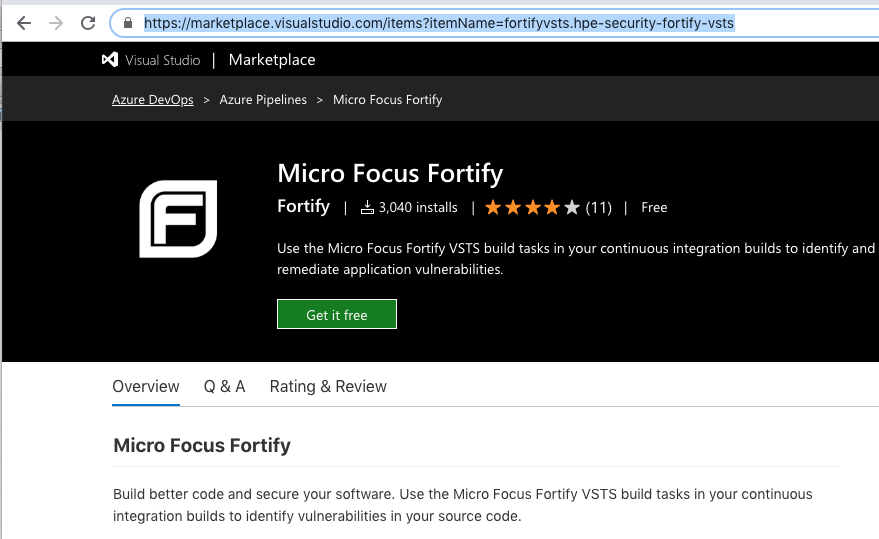
The Microsoft Azure DevOps (VSTS) allows for the building of Java and other languages.

The URL is: https://azure.microsoft.com/en-us/solutions/devops/

## Fortify SCA Visual Studio Plugin on the Microsoft TFS/Azure DevOps Marketplace

**Fortify TFS/Azure DevOps - Plugin URL:**

<https://marketplace.visualstudio.com/items?itemName=fortifyvsts.hpe-security-fortify-vsts>



The Azure DevOps system has a number of tasks that can be used to get all of the source code and Jar dependency files needed for a Java translation. The key items needed for a Java translation is source code and the CLASSPATH set to all required dependency JARs.

The example on the following pages is a simple example of how to scan a Java application.

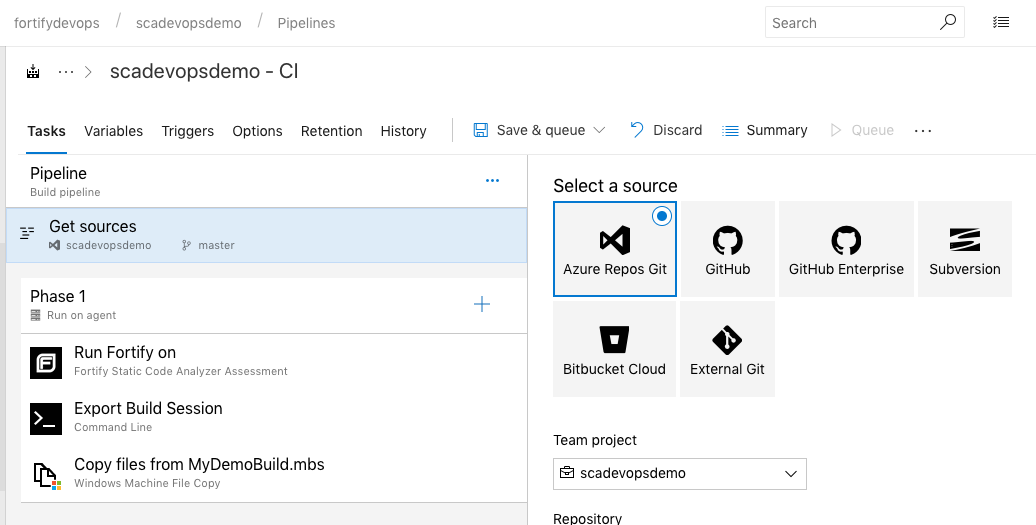
## Translating Java Applications using TFS/Azure DevOps Tasks Plugin

The following images show the Azure DevOps Tasks Options screen for a simple Java project Fortify Translation

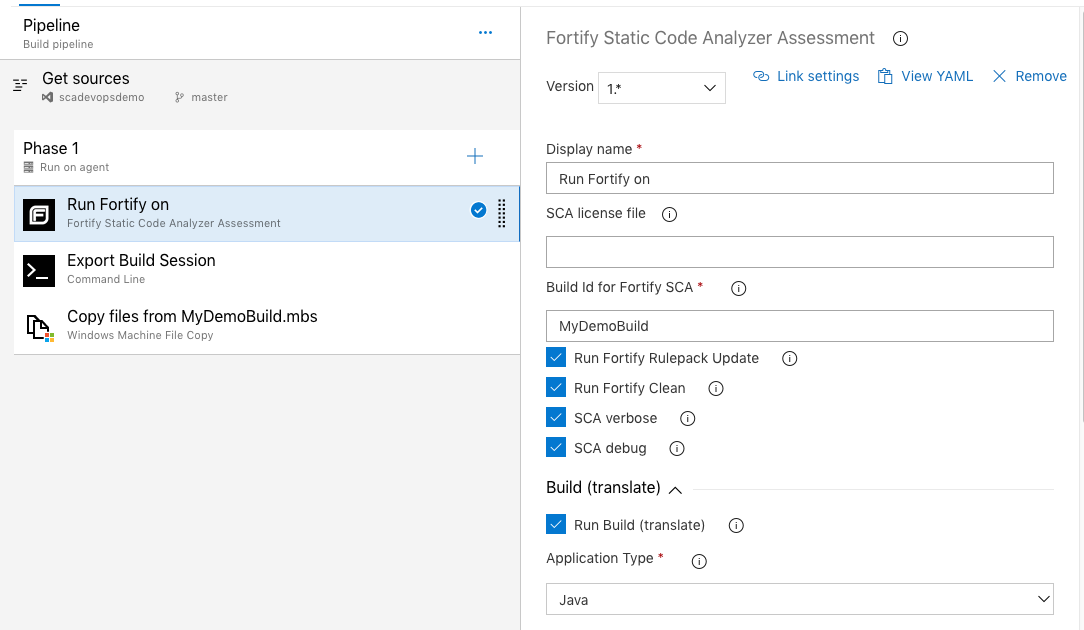
The Steps are: 1 - Checkout Source, 2 - Translate Java, 3 - Create Mobile Build Session (MBS) File, and then 4 - Copy MBS File to Scan machine.

Task 1. ) Check Out source Code and Dependency Jars

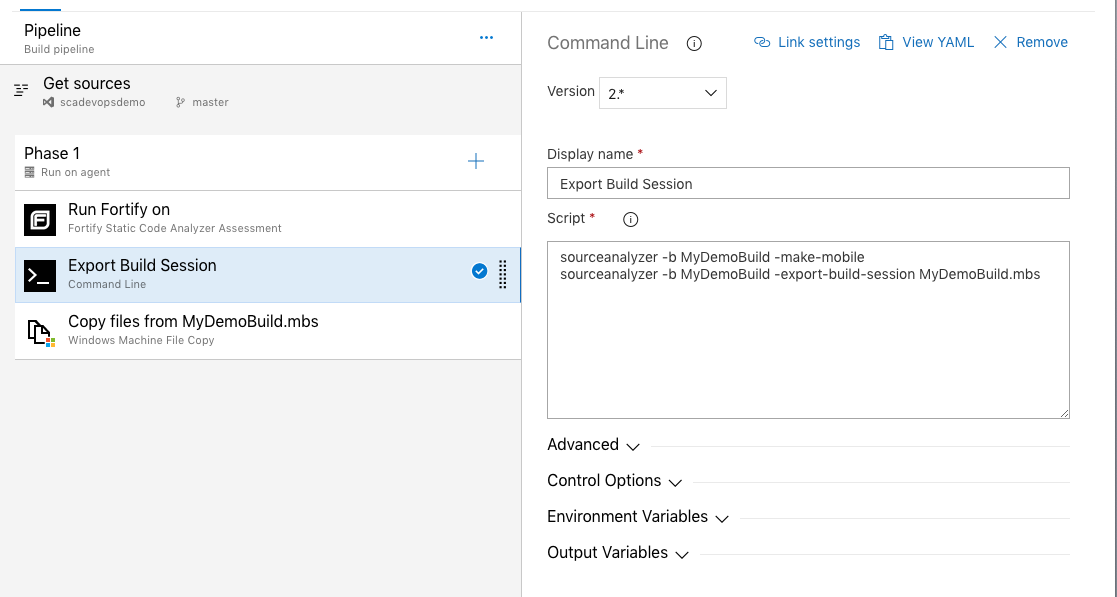
NOTE: You can get source from a variety of sources.



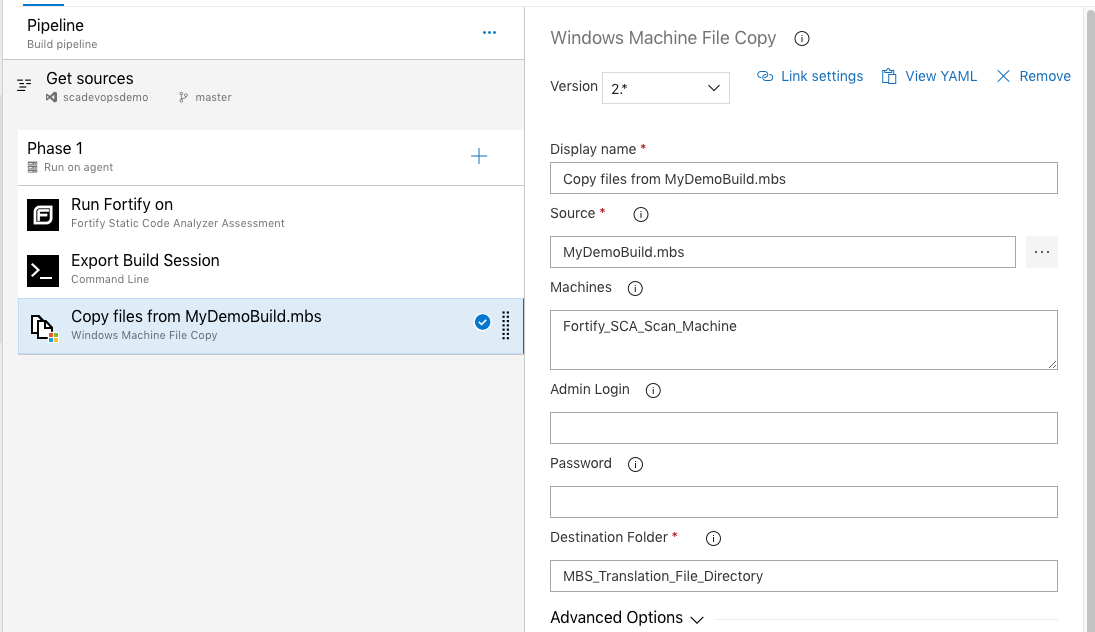
**Task 2. ) Fortify Translate**



**Task 3. ) Create Fortify Mobile Build Session (MBS)**

****

**Task 4. ) Copy MBS to Fortify Scan Machine**

****

# SCA Translation Environment Variables

SCA may use certain environment variables for compiled and build tools. The Build commands have certain environment variable requirements that SCA may or may not use. These will be described in the section on translation by the language type i.e “(Java, Maven, etc.)

## CLASSPATH Environment Variable

SCA will use the environment variable CLASSPATH to resolve the JAR file references in JAVA and JSP translations.

# Changing the default SCA Java JRE in use.

You can remove the SCA OpenJDK Java and specify a JAVA to use that is already installed. To use your installed JAVA, add the environment variable JAVA\_HOME. NOTE: You must the supported Java Version Java 1.8.x.

NOTE: Fortify includes the OpenJDK 1.8 when it is installed. You can rename the SCA Installed JRE directory or remove if you want to use the local version of java that may include security updates. This SCA JRE directory is found in your installed location.

## Linux/Unix/Mac OS Example

"C:\Program Files\Fortify\Fortify\_SCA\_and\_Apps\_18.20\jre\bin\java.exe" -version

openjdk version "1.8.0\_181"

OpenJDK Runtime Environment (Zulu 8.31.0.1-win64) (build 1.8.0\_181-b02)

OpenJDK 64-Bit Server VM (Zulu 8.31.0.1-win64) (build 25.181-b02, mixed mode)

## Example Windows:

set JAVA\_HOME=C:\Program Files\Java\jdk1.8.0\_131

Example Unix/Linux

EXPORT JAVA\_HOME=C:\Program Files\Java\jdk1.8.0\_131

# Visual Studio Translation Basics

* SCA will translate successfully when you have all the Visual Studio Source code (C#,VB, C/C++,etc..) and do not have to reference custom external DLL files.
* SCA will translate successfully on the command line when you use the Visual Studio Solution File and Fortify SCA VS Plugin.
* SCA will translate correctly if the DLLs, PDBs for Source Code and configure the Translate Arguments listed next to point to all the referenced required files, dlls built in DEBUG modes.

## List of arguments for command line without plugin.

* "-dotnet-assembly-name"
* "-dotnet-preproc-symbols"
* "DEBUG;TRACE"
* "-cs-extern-alias"
* "-dotnetwebroot"
* "-libdirs-only"
* "-libdirs"
* "-dotnet-applibs"
* "-dotnet-output-dir"
* "-dotnet-codebehind"

# Recommended Install SCA VS Plugin on All SCA Translation Build systems.

It is recommended, but not required to install the SCA VS Plugin. The SCA VS Plugin will configure the above argument for you when you specify a solution (sln) file. See example on next page.

# Errors in Translation.

SCA will not give an error if the SCA VS plugin is not installed or installed incorrectly. The translation will not be complete if all the required DLL or Source Code files are are not specified. There are no errors related to missing references.

There may be missing file references that mat indicate a DLL is missing. I

It is recommended that you verify that a Solution file compiles correctly before translation using a clean non developer system with just Visual Studio installed and check out from source code system. This will verify that any third party or externally referenced files are available in a plain Visual Studio, otherwise you need to duplicate the developer system on all the build systems.

# SCA will translate successfully for the following versions:

* .NET 2.0 up to .Net 4.7.2
* .NET Core 1.x, 2.0, 2.1
* ASP.NET 2.0 up to ASP.Net 4.7
* C# 5, 6, and 7

Azure DevOps TFS Build Example:

<https://dev.azure.com/scademo/>

# Visual Studio Translation Arguments File Example:

The following is an example of a simple Argument File created by the Visual Studio IDE Plugin.

NOTE: The DLL references and the code behind references below are required to ensure all of the pre-compile translation is complete.

The Solution file for each project and a file with all the references is created like the following below.

When you run a scan in Visual Studio or use the MSBUILD or DEVENV command on the command line with a Solution file.

This file is located in the Directory:

C:\Users\fortify\AppData\Local\Fortify\MSBuildPlugin\<Solution Name>

**SCA Translation VS Plugin Example Arguments File:**

"-machine-output"

"-b"

"DemoDotNet3.sln"

"-dotnet-assembly-name"

"DemoDotNet3"

"-dotnet-preproc-symbols"

"DEBUG;TRACE"

"-cs-extern-alias"

"global=C:\Users\fortify\source\repos\visualStudiotips\ExternalLib\lib\ExternalLib.dll,global=C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.CodeDom.Providers.DotNetCompilerPlatform.2.0.0\lib\net45\Microsoft.CodeDom.Providers.DotNetCompilerPlatform.dll,global=C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\Microsoft.CSharp.dll,global=C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.Web.Infrastructure.1.0.0.0\lib\net40\Microsoft.Web.Infrastructure.dll,global=C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\mscorlib.dll,global=C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Newtonsoft.Json.11.0.1\lib\net45\Newtonsoft.Json.dll,global=C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.ComponentModel.DataAnnotations.dll,global=C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Configuration.dll,global=C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Core.dll,global=C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Data.DataSetExtensions.dll,global=C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Data.dll,global=C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.dll,global=C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Drawing.dll,global=C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.EnterpriseServices.dll,global=C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Net.Http.dll,global=C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.AspNet.WebApi.Client.5.2.4\lib\net45\System.Net.Http.Formatting.dll,global=C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Web.ApplicationServices.dll,global=C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Web.dll,global=C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Web.DynamicData.dll,global=C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Web.Entity.dll,global=C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Web.Extensions.dll,global=C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.AspNet.Webpages.3.2.4\lib\net45\System.Web.Helpers.dll,global=C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.AspNet.WebApi.Core.5.2.4\lib\net45\System.Web.Http.dll,global=C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.AspNet.WebApi.WebHost.5.2.4\lib\net45\System.Web.Http.WebHost.dll,global=C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.AspNet.Mvc.5.2.4\lib\net45\System.Web.Mvc.dll,global=C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.AspNet.Razor.3.2.4\lib\net45\System.Web.Razor.dll,global=C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Web.Services.dll,global=C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.AspNet.Webpages.3.2.4\lib\net45\System.Web.Webpages.Deployment.dll,global=C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.AspNet.Webpages.3.2.4\lib\net45\System.Web.Webpages.dll,global=C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.AspNet.Webpages.3.2.4\lib\net45\System.Web.Webpages.Razor.dll,global=C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Xml.dll,global=C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Xml.Linq.dll"

"-dotnetwebroot"

"C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\\"

"-libdirs-only"

"-libdirs"

"C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\Microsoft.CSharp.dll;C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\mscorlib.dll;C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.ComponentModel.DataAnnotations.dll;C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Configuration.dll;C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Core.dll;C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Data.DataSetExtensions.dll;C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Data.dll;C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.dll;C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Drawing.dll;C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.EnterpriseServices.dll;C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Net.Http.dll;C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Web.ApplicationServices.dll;C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Web.dll;C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Web.DynamicData.dll;C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Web.Entity.dll;C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Web.Extensions.dll;C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Web.Services.dll;C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Xml.dll;C:\Program Files (x86)\Reference Assemblies\Microsoft\Framework\.NETFramework\v4.7.2\System.Xml.Linq.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.AspNet.Mvc.5.2.4\lib\net45\System.Web.Mvc.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.AspNet.Razor.3.2.4\lib\net45\System.Web.Razor.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.AspNet.WebApi.Client.5.2.4\lib\net45\System.Net.Http.Formatting.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.AspNet.WebApi.Core.5.2.4\lib\net45\System.Web.Http.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.AspNet.WebApi.WebHost.5.2.4\lib\net45\System.Web.Http.WebHost.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.AspNet.Webpages.3.2.4\lib\net45\System.Web.Helpers.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.AspNet.Webpages.3.2.4\lib\net45\System.Web.Webpages.Deployment.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.AspNet.Webpages.3.2.4\lib\net45\System.Web.Webpages.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.AspNet.Webpages.3.2.4\lib\net45\System.Web.Webpages.Razor.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.CodeDom.Providers.DotNetCompilerPlatform.2.0.0\lib\net45\Microsoft.CodeDom.Providers.DotNetCompilerPlatform.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Microsoft.Web.Infrastructure.1.0.0.0\lib\net40\Microsoft.Web.Infrastructure.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\packages\Newtonsoft.Json.11.0.1\lib\net45\Newtonsoft.Json.dll;C:\Users\fortify\source\repos\visualStudiotips\ExternalLib\lib\ExternalLib.dll"

"-dotnet-applibs"

"C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\DemoDotNet3.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\ExternalLib.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\Microsoft.CodeDom.Providers.DotNetCompilerPlatform.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\Microsoft.Web.Infrastructure.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\Newtonsoft.Json.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\System.Net.Http.Formatting.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\System.Web.Helpers.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\System.Web.Http.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\System.Web.Http.WebHost.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\System.Web.Mvc.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\System.Web.Razor.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\System.Web.Webpages.Deployment.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\System.Web.Webpages.dll;C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\System.Web.Webpages.Razor.dll"

"-dotnet-output-dir"

"C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\\"

"-dotnet-codebehind"

"c:\users\fortify\source\repos\visualstudiotips\demodotnet3\demodotnet3\global.asax.cs"

"c:\users\fortify\source\repos\visualstudiotips\demodotnet3\demodotnet3\app\_start\routeconfig.cs"

"c:\users\fortify\source\repos\visualstudiotips\demodotnet3\demodotnet3\app\_start\webapiconfig.cs"

"c:\users\fortify\source\repos\visualstudiotips\demodotnet3\demodotnet3\externalclass1.cs"

"c:\users\fortify\source\repos\visualstudiotips\demodotnet3\demodotnet3\properties\assemblyinfo.cs"

"c:\users\fortify\appdata\local\temp\.netframework,version=v4.7.2.assemblyattributes.cs"

"c:\users\fortify\source\repos\visualstudiotips\demodotnet3\demodotnet3\global.asax"

"c:\users\fortify\source\repos\visualstudiotips\demodotnet3\demodotnet3\web.config"

"c:\users\fortify\source\repos\visualstudiotips\demodotnet3\demodotnet3\views\web.config"

"c:\users\fortify\source\repos\visualstudiotips\demodotnet3\demodotnet3\packages.config"

"c:\users\fortify\source\repos\visualstudiotips\demodotnet3\demodotnet3\web.debug.config"

"c:\users\fortify\source\repos\visualstudiotips\demodotnet3\demodotnet3\web.release.config"

"C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\DemoDotNet3.dll.config"

"C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\Microsoft.CodeDom.Providers.DotNetCompilerPlatform.xml"

"C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\Newtonsoft.Json.xml"

"C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\System.Net.Http.Formatting.xml"

"C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\System.Web.Helpers.xml"

"C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\System.Web.Http.WebHost.xml"

"C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\System.Web.Http.xml"

"C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\System.Web.Mvc.xml"

"C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\System.Web.Razor.xml"

"C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\System.Web.Webpages.Deployment.xml"

"C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\System.Web.Webpages.Razor.xml"

"C:\Users\fortify\source\repos\visualStudiotips\DemoDotNet3\DemoDotNet3\bin\System.Web.Webpages.xml"

# Appendix

## Example Verify a Translation Script.

#

# Find all Files that could be translated by Fortify

#

# Description

# This Script will read the fortify sca properties file and find all the supported types and

# Search the directory for all files that match supported file types by extension and

# compare against the Build ID List of file translated

#

function checktranslation()

{

#echo "DEBUG: checking: $fileext"

for filesrc in `find . -name "\*${fileext}"`

do

basepath=`echo $filesrc | cut -c3-`

grep $basepath ${TMPFILE} > foo

if [ $? = 0 ];

then

foo=foo

else

echo "SCA Source File Not Translates: ($filesrc)"

fi

done

}

TMPFILE="`pwd`/scaTranslated.tmp"

SCACMD=`which sourceanalyzer `

SCAConfigDir=`echo ${SCACMD} | sed 's;bin/sourceanalyzer;Core/config;'`

FortifyConfigFile="${SCAConfigDir}/fortify-sca.properties"

BUILDID=$1

SOURCEDIR=$2

if [ $# -gt 1 ];

then

if [ ! -d ${SOURCEDIR} ];

then

echo "ERROR: Source not found ! Verify it exists and is readable"

exit 1

fi

if [ -r "$FortifyConfigFile" ];

then

$SCACMD -b ${BUILDID} -show-files > ${TMPFILE}

cd ${SOURCEDIR}

# Set File Extension Type for supported Translation languages.

scafileext=`grep com.fortify.sca.DefaultFileTypes $FortifyConfigFile`

cnt=1

fileext=`echo ${scafileext} | cut -d= -f2 | cut -d, -f1 `

while true

do

cnt=`expr $cnt + 1`

if [ "$fileext" != "" ];

then

# Find All potential files that can be translated

checktranslation

fileext=`echo ${scafileext} | cut -d\, -f${cnt}`

else

exit 0

fi

done

else

echo "ERROR: Fortify Config File not Found at: ($FortifyConfigFile)"

echo "Verify that sourceanalyzer is install and in your PATH . "

exit 2

fi

else

echo "Usage: verifyTranalation.sh <SCA BUILD ID> <Source Code Dir>"

exit 99

fi

## Example Simple Java Translation Script

#

# This Simple Script to translate java source code by creating the CLASSPATH from all

# JARs files found in the source code directory tree.

#

#

SCAVALID=`which sourceanalyzer`

if [ $? -ne 0 ];

then

echo "ERROR: sourceanalyzer is missing from your bin PATH directories."

echo "FIX: set the PATH to the location of the installed bin directory for sourceanalyzer"

exit 99

fi

if [ $# -gt 1 ];

then

BUILD\_ID=$1

SOURCE\_DIR=$2

CLASSPATH=${SOURCE\_DIR}:

for jarfile in `find ${SOURCE\_DIR} -name "\*.jar" `

do

CLASSPATH="${CLASSPATH}:${jarfile}:"

done

sourceanalyzer -b ${BUILD\_ID} -clean

sourceanalyzer -b ${BUILD\_ID} -verbose -logfile ${BUILD\_ID}.log -debug -cp ${CLASSPATH} ${SOURCE\_DIR}

sourceanalyzer -b ${BUILD\_ID} -make-mobile

sourceanalyzer -b ${BUILD\_ID} -export-build-session ${BUILD\_ID}.mbs

else

echo "usage translateJava.sh <BUILD ID> <Source Directory>"

exit 1

fi