Final_Automation_Code.sh

The script you are about to see is created for your regular ActiveMQ project build failures due common issue in the dependencies, which includes setting up Java environment properly and checking out a particular commit where needed followed by applying fixes in Maven dependency error commonly occurred while building this project. The script line-by-line breakdown:

Script Header and Input Variables

```
#!/bin/bash
# Script to automatically fix build dependency errors for ActiveMQ project
# Takes commit_ID, module_name, and Java version as input
commit_ID=$1
module_name=$2
java_version=$3
branch_name="temp61001_fix_branch"
original_pom="pom.xml"
backup_pom="pom_backup.xml"
```

- #!/bin/bash defines the script as a Bash script.
- Comments provide an overview.
- The script expects three command-line arguments: commit_ID, module_name, and java_version.
- Defines a branch name and pom.xml file paths for easy reference.

Function Definitions

1. **Setting Java Version**

```
set_java_version() {
    echo "Setting Java version to $java_version..."
    case "$java_version" in
        "8")
            export JAVA_HOME="/c/Program Files/Java/jdk1.8.0_202"
            ;;
        "11")
            export JAVA_HOME="/c/Program Files/Java/jdk-11"
            ;;
        "17")
            export JAVA_HOME="/c/Program Files/Java/jdk-17"
            ;;
        "20")
            export JAVA_HOME="/c/Program Files/Java/jdk-20"
            ;;
        *)
            echo "Unsupported Java version specified. Please use 8, 11, 17, or 20."
            exit 1
            ;;
    esac
    export PATH="$JAVA_HOME/bin:$PATH"
    echo "JAVA_HOME set to $JAVA_HOME"
    echo "Java version set to $("$JAVA_HOME/bin/java" -version 2>&1 | head -n 1)"
}
```

- Sets the JAVA_HOME environment variable based on java_version specified.
- Updates the PATH to prioritize this Java version.
- Provides feedback by displaying the selected Java version.

2. Checking Out the Commit and Creating a Branch

```
checkout_commit() {

    echo "Checking out commit: $commit_ID"
    git checkout $commit_ID
    if [ $? -ne 0 ]; then
        echo "Failed to checkout commit $commit_ID"
        exit 1
    fi
    echo "Creating a new temporary branch: $branch_name"
    git checkout -b $branch_name
    if [ $? -ne 0 ]; then
        echo "Failed to create branch $branch_name"
        exit 1
    fi
}
```

- Checks out the specified Git commit, creating a new branch to isolate changes.
- Exits with an error if either operation fails.

3. Backup and Restore pom.xml

```
backup_pom() {
    if [ -f "$original_pom" ]; then
        echo "Backing up original pom.xml to $backup_pom"
        cp "$original_pom" "$backup_pom"
    fi
}

restore_pom() {
    if [ -f "$backup_pom" ]; then
        echo "Restoring the original pom.xml"
        mv "$backup_pom" "$original_pom"
    fi
}
```

- backup_pom copies pom.xml to a backup file if it exists.
- restore_pom restores pom.xml from the backup, ensuring no permanent modifications.

4. Fixing Dependency Issues

```
fix_non_resolvable_parent_pom() {
    echo "Attempting to fix non-resolvable parent POM issue..."
    mvn clean install -U -DskipTests
    if ! mvn validate; then
        echo "Parent POM not found. Trying to install manually..."
    fi
}
```

- Performs a forced Maven update (mvn clean install -U) to resolve dependencies.
- Attempts mvn validate to confirm dependencies are available locally.

Missing activemq-ra Dependency

```
fix_activemq_ra_dependency() {
    echo "Fixing missing activemq-ra dependency..."
    if ! grep -q "<id>apache.snapshots</id>" "$original_pom"; then
        echo "Adding Apache's snapshot repository to pom.xml..."
        sed -i '/<\/project>/i\
<repositories>\
    <repository>\
        <id>apache.snapshots</id>\
        <url>https://repository.apache.org/content/repositories/snapshots/</url>\
        <snapshots>\
            <enabled>true</enabled>\
        </snapshots>\
    </repository>\
</repositories>' "$original_pom"
    fi
    echo "Updating activemq-ra dependency to stable version..."
    sed -i '/<artifactId>activemq-ra<\/artifactId>/!b;n;c\        <version>5.4.0<\/version>'
"$original_pom"
}
```

- Checks for Apache snapshot repository in pom.xml and adds it if missing.
- Updates the activemq-ra dependency to a stable version.

Fixing xbean-spring and Other Dependencies

- Each of these follows a similar pattern:

- Adds the Apache snapshot repository if absent.
- Updates the specified dependency version to a stable version.

5. Detecting and Fixing Duplicated `<repositories>` Tags

```
fix_duplicated_repositories_tag() {
   duplicated_tag_count=$(grep -c "<repositories>" "$original_pom")
   if [ "$duplicated_tag_count" -gt 1 ]; then
      first_repo_position=$(grep -n "<repositories>" "$original_pom" | head -n 1 | cut -d: -f1)
      sed -i "$(("$first_repo_position" + 1)),\$s/<repositories>.*<\/repositories>/<!-- Duplicated repositories removed -->/" "$original_pom"
   else
      echo "No duplicated <repositories> tag found."
   fi
}
```

- Counts <repositories> tags and removes duplicates if present, avoiding Maven build conflicts.

6. Compiling the Module

```
compile_module() {
   mvn clean install -DskipTests -pl $module_name -am
   if [ $? -eq 0 ]; then
      echo "Build succeeded!"
      return 0
   else
      echo "Build failed. Checking errors..."
      return 1
   fi
}
```

- Compiles the specified module, skipping tests to focus on dependencies.
- Returns 0 on success, 1 on failure.

7. Cleaning Up the Temporary Branch

```
cleanup_branch() {
   git checkout $commit_ID
   git branch -D $branch_name
}
```

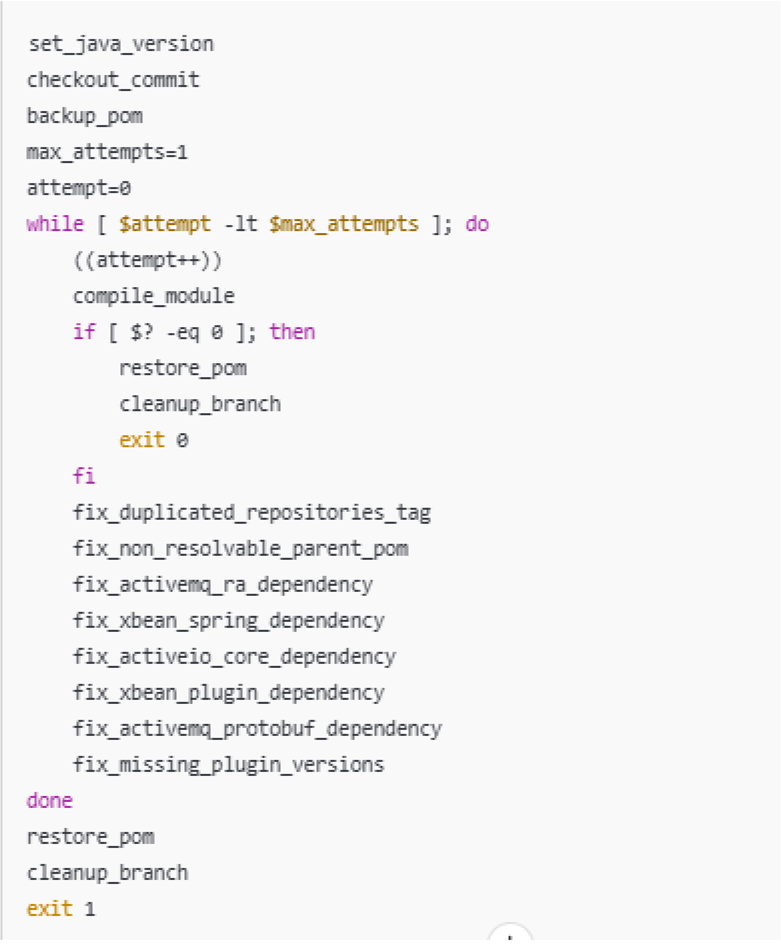
- Deletes the temporary branch after the process.

8. Detecting Missing Plugin Versions

```
fix_missing_plugin_versions() {
   mvn versions:display-plugin-updates | grep -A 2 "Plugin Plugin version"
   mvn versions:use-latest-versions
}
```

- Detects outdated or missing plugin versions and updates them.

Main Script Execution



```
set_java_version
checkout_commit
backup_pom
max_attempts=1
attempt=0
while [ $attempt -lt $max_attempts ]; do
    ((attempt++))
    compile_module
    if [ $? -eq 0 ]; then
        restore_pom
        cleanup_branch
        exit 0
    fi
    fix_duplicated_repositories_tag
    fix_non_resolvable_parent_pom
    fix_activemq_ra_dependency
    fix_xbean_spring_dependency
    fix_activeio_core_dependency
    fix_xbean_plugin_dependency
    fix_activemq_protobuf_dependency
    fix_missing_plugin_versions
done
restore_pom
cleanup_branch
exit 1
```

- ➢ Runs the functions in order, attempting to fix issues and build the module.
- ➢ If successful, it restores pom.xml and deletes the temporary branch. Otherwise, it exits with an error message.