

```

#!/usr/bin/env sh
\\
#####
#####
##
## Gradle start up script for UN*X
##
#####
#####

# Attempt to set APP_HOME
# Resolve links: $0 may be a link
PRG="$0"
# Need this for relative symlinks.
while [ -h "$PRG" ]; do
    ls=`ls -ld "$PRG"`
    link=`expr "$ls" : '.*-> \(.*\)$'`
    if expr "$link" : '/.*' > /dev/null; then
        PRG="$link"
    else
        PRG=`dirname "$PRG"`"/$link"
    fi
done
SAVED=""`pwd`"
cd "`dirname \"$PRG\"`/" >/dev/null
APP_HOME=""`pwd -P`"
cd "$SAVED" >/dev/null

APP_NAME="Gradle"
APP_BASE_NAME=`basename "$0"`

# Add default JVM options here. You can also use JAVA_OPTS and GRADLE_OPTS to pass JVM
options to this script.
DEFAULT_JVM_OPTS=""

# Use the maximum available, or set MAX_FD != -1 to use that value.
MAX_FD="maximum"

warn () {
    echo "$*"
}

```

```
die () {  
    echo  
    echo "$*"  
    echo  
    exit 1  
}
```

```
# OS specific support (must be 'true' or 'false').
```

```
cygwin=false  
msys=false  
darwin=false  
nonstop=false  
case "`uname`" in  
    CYGWIN* )  
        cygwin=true  
        ;;  
    Darwin* )  
        darwin=true  
        ;;  
    MINGW* )  
        msys=true  
        ;;  
    NONSTOP* )  
        nonstop=true  
        ;;  
esac
```

```
CLASSPATH=$APP_HOME/gradle/wrapper/gradle-wrapper.jar
```

```
# Determine the Java command to use to start the JVM.
```

```
if [ -n "$JAVA_HOME" ] ; then  
    if [ -x "$JAVA_HOME/jre/sh/java" ] ; then  
        # IBM's JDK on AIX uses strange locations for the executables  
        JAVACMD="$JAVA_HOME/jre/sh/java"  
    else  
        JAVACMD="$JAVA_HOME/bin/java"  
    fi  
    if [ ! -x "$JAVACMD" ] ; then  
        die "ERROR: JAVA_HOME is set to an invalid directory: $JAVA_HOME"
```

Please set the JAVA_HOME variable in your environment to match the location of your Java installation."

```
fi
else
    JAVACMD="java"
    which java >/dev/null 2>&1 || die "ERROR: JAVA_HOME is not set and no 'java' command could
be found in your PATH.
```

Please set the JAVA_HOME variable in your environment to match the location of your Java installation."

```
fi
```

```
# Increase the maximum file descriptors if we can.
```

```
if [ "$cygwin" = "false" -a "$darwin" = "false" -a "$nonstop" = "false" ]; then
    MAX_FD_LIMIT=`ulimit -H -n`
    if [ $? -eq 0 ]; then
        if [ "$MAX_FD" = "maximum" -o "$MAX_FD" = "max" ]; then
            MAX_FD="$MAX_FD_LIMIT"
        fi
        ulimit -n $MAX_FD
        if [ $? -ne 0 ]; then
            warn "Could not set maximum file descriptor limit: $MAX_FD"
        fi
    else
        warn "Could not query maximum file descriptor limit: $MAX_FD_LIMIT"
    fi
fi
```

```
# For Darwin, add options to specify how the application appears in the dock
```

```
if $darwin; then
```

```
    GRADLE_OPTS="$GRADLE_OPTS \"-Xdock:name=$APP_NAME\" \"-
Xdock:icon=$APP_HOME/media/gradle.icns\""
fi
```

```
# For Cygwin, switch paths to Windows format before running java
```

```
if $cygwin ; then
```

```
    APP_HOME=`cygpath --path --mixed "$APP_HOME"`
    CLASSPATH=`cygpath --path --mixed "$CLASSPATH"`
    JAVACMD=`cygpath --unix "$JAVACMD"`
```

```
# We build the pattern for arguments to be converted via cygpath
```

```

ROOTDIRSRAW=`find -L / -maxdepth 1 -mindepth 1 -type d 2>/dev/null`
SEP=""
for dir in $ROOTDIRSRAW ; do
    ROOTDIRS="$ROOTDIRS$SEP$dir"
    SEP="|"
done
OURCYGPATTERN="^(($ROOTDIRS))"
# Add a user-defined pattern to the cygpath arguments
if [ "$GRADLE_CYGPATTERN" != "" ] ; then
    OURCYGPATTERN="$OURCYGPATTERN|($GRADLE_CYGPATTERN)"
fi
# Now convert the arguments - kludge to limit ourselves to /bin/sh
i=0
for arg in "$@" ; do
    CHECK=`echo "$arg"|egrep -c "$OURCYGPATTERN" -`
    CHECK2=`echo "$arg"|egrep -c "^-"`           ### Determine if an option

    if [ $CHECK -ne 0 ] && [ $CHECK2 -eq 0 ] ; then        ### Added a condition
        eval `echo args$i`=`cygpath --path --ignore --mixed "$arg"`
    else
        eval `echo args$i`="\"$arg\""
    fi
    i=$((i+1))
done
case $i in
    (0) set -- ;;
    (1) set -- "$args0" ;;
    (2) set -- "$args0" "$args1" ;;
    (3) set -- "$args0" "$args1" "$args2" ;;
    (4) set -- "$args0" "$args1" "$args2" "$args3" ;;
    (5) set -- "$args0" "$args1" "$args2" "$args3" "$args4" ;;
    (6) set -- "$args0" "$args1" "$args2" "$args3" "$args4" "$args5" ;;
    (7) set -- "$args0" "$args1" "$args2" "$args3" "$args4" "$args5" "$args6" ;;
    (8) set -- "$args0" "$args1" "$args2" "$args3" "$args4" "$args5" "$args6" "$args7" ;;
    (9) set -- "$args0" "$args1" "$args2" "$args3" "$args4" "$args5" "$args6" "$args7" "$args8" ;;
esac
fi

# Escape application args
save () {
    for i do printf %s\\n "$i" | sed "s/'/'\\\\"/g;s/^/\\/;\\$s/\\$/ '\\\\V"; done

```

```
    echo " "
}
APP_ARGS=$(save "$@")

# Collect all arguments for the java command, following the shell quoting and substitution rules
eval set -- $DEFAULT_JVM_OPTS $JAVA_OPTS $GRADLE_OPTS "\-
Dorg.gradle.appname=$APP_BASE_NAME\\"" -classpath "\"$CLASSPATH\\""
org.gradle.wrapper.GradleWrapperMain "$APP_ARGS"

# by default we should be in the correct project dir, but when run from Finder on Mac, the cwd is
wrong
if [ "$(uname)" = "Darwin" ] && [ "$HOME" = "$PWD" ]; then
    cd "$(dirname "$0")"
fi

exec "$JAVACMD" "$@"
```