

```
#!/bin/bash
```

```
# Copyright (c) 2021-2023 by Ivanti, Inc. All rights reserved
```

```
#set -x
```

```
PLATFORM=`/bin/uname`
```

```
ARCH=`/bin/uname -m`
```

```
# All disk space in MegaBytes
```

```
CEF_DISPLAY_NAME="Chromium Embedded Browser"
```

```
DF=$(which df)
```

```
DATE=$(which date)
```

```
RM=$(which rm)
```

```
CP=$(which cp)
```

```
MKDIR=$(which mkdir)
```

```
ECHO=$(which echo)
```

```
CURL=$(which curl)
```

```
WGET=$(which wget)
```

```
UNZIP=$(which unzip)
```

```
GREP=$(which grep)
```

```
AWK=$(which awk)
```

```
TAR=$(which tar)
```

```
CUT=$(which cut)
```

```
SHASUM=$(which shasum)
```

```
if [ "$PLATFORM" == "Linux" ]; then
```

```
    TMP_SPACE_NEEDED=1276
```

```
    INSTALL_SPACE_NEEDED=1063
```

```
    CEF_INSTALL_ROOT_DIR=/opt
```

```
    CEF_INSTALL_DIR=${CEF_INSTALL_ROOT_DIR}/pulsesecure/lib/cefRuntime
```

```
    CEF_LIB_NAME=libcef.so
```

```
    CEF_LIB_SHA2=cdd94ba80748e78118835321bbd5fc8be23b03e9c11cab083640c3f558a79d01
```

```
    CEF_URL=https://pcstp.pulsesecure.net/cef/linux/cef64.94.tar.bz2
```

```
    CEF_PACKAGE_SHA2=0612d5b88b83d3283c9b5ade6e87295df103def771bad1ad32aea6adf42dc6e3
```

```
    CEF_CHECK_INSTALL_PATH=${CEF_INSTALL_DIR}/Release/${CEF_LIB_NAME}
```

```
    CEF_DOWNLOAD_FILE=cef64.94.tar.bz2
```

```
    CEF_PACKAGE_NAME=cef_binary_94.4.8+g5b52963+chromium-94.0.4606.71_linux64_minimal
```

```
    CEF_RESOURCE_DIR=${CEF_PACKAGE_NAME}/Resources
```

```
    CEF_RELEASE_DIR=${CEF_PACKAGE_NAME}/Release
```

```
    PULSE_LOG_DIR="/var/log/pulsesecure/logging"
```

```
else
```

```
    TMP_SPACE_NEEDED=300
```

```
    INSTALL_SPACE_NEEDED=200
```

```
    CEF_INSTALL_ROOT_DIR=/Library
```

```
    CEF_DIRNAME="Chromium Embedded Framework.framework"
```

```
    CEF_INSTALL_DIR2="${CEF_INSTALL_ROOT_DIR}/Application Support/Pulse Secure"
```

```
    CEF_INSTALL_DIR="${CEF_INSTALL_DIR2}/${CEF_DIRNAME}"
```

```
    CEF_LIB_NAME="Chromium Embedded Framework"
```

```
if [ "$ARCH" == "x86_64" ]; then
```

```
    CEF_URL=https://pcstp.pulsesecure.net/cef/macOS/cef64.92.zip
```

```
    CEF_PACKAGE_SHA2=f32923ff9c0415efe47300a8d40e37ab1738833526529da77ae8e086c6b14811
```

```
    CEF_LIB_SHA2=5cdb3b860681d260c2ac67c03a45930db4811b79ec8c4d00b14ce89f281705e5
```

```
else
```

```
    CEF_URL=https://pcstp.pulsesecure.net/cef/macOS/cefarm64.92.zip
```

```
CEF_PACKAGE_SHA2=491cb7e770852d49e66647e833cad3d7db8e6c871b5185df373e6354be2dd1a7  
CEF_LIB_SHA2=5a0ad53af6db575510e9e9045765d7643e915eda677c50c3911b4a1e22508c57
```

```
fi
```

```
CEF_CHECK_INSTALL_PATH="$CEF_INSTALL_DIR/$CEF_LIB_NAME"  
CEF_DOWNLOAD_FILE=cef64.zip  
PULSE_LOG_DIR="/var/log/Pulse Secure/Logging"
```

```
fi
```

```
TMP_DIR=/tmp/cef.download  
REINSTALL_CEF=false
```

```
LOG_DIR=$PULSE_LOG_DIR  
TMP_DIR_PARAM=tmpDirPath  
SCRNAME=`basename $0`  
attempting_copy=false
```

function printUsage()

```
{
$ECHO "Utility to download and install $CEF_DISPLAY_NAME."
$ECHO "Usage: $SCRNAME <check_installed|install|reinstall|uninstall> [OPTIONS]"
$ECHO ""
$ECHO "  $SCRNAME check_installed"
$ECHO "    Checks and informs if currently installed or not."
$ECHO "  $SCRNAME <install|reinstall> [-$TMP_DIR_PARAM <Path>]"
$ECHO "    Downloads and installs the $CEF_DISPLAY_NAME."
$ECHO "    An optional -$TMP_DIR_PARAM can be passed, where download and"
$ECHO "    extraction will happen. This directory will be cleared by the"
$ECHO "    end of installation."
$ECHO "    install will install CEF only if it is not installed."
$ECHO "    reinstall will remove and reinstall if it is installed."
$ECHO "  $SCRNAME uninstall"
$ECHO "    Uninstall CEF libraries"
$ECHO "  Note:"
$ECHO "    * The install can be run only with root privileges"
$ECHO "    * $TMP_SPACE_NEEDED MB of free space is needed in the"
$ECHO "    $TMP_DIR_PARAM. This space is used only during installation"
$ECHO "    time and freed by the end of installation."
$ECHO "    * $INSTALL_SPACE_NEEDED MB of free space is needed in the ${CEF_INSTALL_ROOT_DIR}"
exit 1
}
```

function get_time()

```
{  
  $DATE '+%Y/%m/%d %H:%M:%S.%N'  
}
```

function log_message()

```
{  
  LOG_FILE="$LOG_DIR/cefinstaller.log"  
  msg=$1  
  $ECHO $msg  
  $ECHO "$(get_time) $msg" >> "$LOG_FILE"  
}
```

ALREADY_INSTALLED=false

function check_installed()

```
{  
  if [ -f "$CEF_CHECK_INSTALL_PATH" ]; then  
    log_message "$CEF_DISPLAY_NAME is installed"  
    validate_lib_checksum  
    ALREADY_INSTALLED=true  
  else  
    log_message "$CEF_DISPLAY_NAME is not installed"  
  fi  
}
```

function get_root_dir()

```
{  
  $ECHO "$1" | $CUT -d "/" -f2  
}
```

function get_file_system()

```
{  
  $DF $1 | $GREP -vw Filesystem | $AWK {'print $1'}  
}
```

function get_free_space()

```
{  
  $DF -m $1 | $GREP -vw Filesystem | $AWK {'print $4'}  
}
```

function get_space_util()

```
{  
  $DF -m $1 | $GREP -vw Filesystem | $AWK {'print $5'}  
}
```

function get_fs_desc()

```
{  
  $DF -m $1 | $GREP -vw Filesystem | $AWK {'printf "%s(%s)", $6, $1'}  
}
```

function get_space_message()

```
{  
  $ECHO "Not enough free disk space on $(get_fs_desc $1). Needed:$2MB Available:$3MB"  
}
```

function check_partition_space()

```
{
  dirPath=$1
  freespaceneeded=$2
  freediskspace=$(get_free_space $dirPath)
  if [ $freediskspace -le $freespaceneeded ]; then
    log_message "$(get_space_message $dirPath $freespaceneeded $freediskspace)"
    exit 2
  fi
}
```

function check_root_privilege()

```
{
  if [ $EUID -ne 0 ]; then
    log_message "'$SCRNAME install|reinstall|uninstall' needs to be run with root privileges"
    exit 3
  fi
}
```

function check_disk_space()

```
{
  log_message "Checking disk space needed..."
  if [ ! -d "$TMP_DIR" ]; then
    $MKDIR -p "$TMP_DIR"
  fi
  tmpPartition=$(get_file_system $TMP_DIR)
  cefInstallRoot=$(get_root_dir $CEF_INSTALL_DIR)
  installPartition=$(get_file_system $cefInstallRoot)
  if [ "$tmpPartition" == "$installPartition" ]; then
    check_partition_space $cefInstallRoot $(( $TMP_SPACE_NEEDED + $INSTALL_SPACE_NEEDED ))
  else
    check_partition_space $cefInstallRoot $INSTALL_SPACE_NEEDED
    check_partition_space $TMP_DIR $TMP_SPACE_NEEDED
  fi
  log_message "Disk space available"
}
```

function return_error()

```
{
  exitValue=$1
  errorMessage=$2

  if [ "100%" == "$(get_space_util $TMP_DIR)" ]; then
    log_message "Warning: $TMP_DIR is full"
  fi

  log_message "$errorMessage"
  if [ $exitValue -ne 0 ]; then
    cleanup
  fi
  exit $exitValue
}
```

function cleanup()

```
{
  log_message "Cleaning up..."
  if [ -d $TMP_DIR/$CEF_PACKAGE_NAME ]; then
    $RM -rf "$TMP_DIR/$CEF_PACKAGE_NAME"
  fi
  if [ -f $TMP_DIR/$CEF_DOWNLOAD_FILE ]; then
    $RM -rf "$TMP_DIR/$CEF_DOWNLOAD_FILE"
  fi
  if [ "$attempting_copy" == "true" ]; then
    $RM -rf "$CEF_INSTALL_DIR/*"
  fi
  log_message "Cleanup done"
}
```

function download_cef()

```
{
  log_message "Downloading CEF from $CEF_URL..."

  RETRIES=3
  if [ -f $CURL ]; then
    $CURL --retry $RETRIES -C - $CEF_URL -o $TMP_DIR/$CEF_DOWNLOAD_FILE --connect-timeout 30
  elif [ -f $WGET ]; then
    $WGET --tries $RETRIES --continue --show-progress $CEF_URL --directory-prefix=$TMP_DIR --connect-
timeout=30
  else
    return_error 8 "Error: curl or wget are needed for downloading."
  fi

  if [ $? -ne 0 ]; then
    return_error 4 "Error: Download failed. Please try again."
  fi
  log_message "Download successful"
}
```

function validate_lib_checksum()

```
{
  log_message "Validating the CEF lib..."
  sha2checksum=`$SHASUM -a 256 "$CEF_CHECK_INSTALL_PATH" | $CUT -d' ' -f1`
  if [ "$sha2checksum" != "$CEF_LIB_SHA2" ]; then
    log_message "$CEF_DISPLAY_NAME latest version is not installed"
    exit 5
  fi
  log_message "CEF lib Validation successful"
}
```

function validate_checksum()

```
{
  log_message "Validating the downloaded package..."
  sha2checksum=`$SHASUM -a 256 $TMP_DIR/$CEF_DOWNLOAD_FILE | $CUT -d' ' -f1`
  if [ "$sha2checksum" != "$CEF_PACKAGE_SHA2" ]; then
    return_error 5 "Error: Download package file verification failed. Please try again"
  fi
  log_message "Validation successful"
}
```

function extract_cef()

```
{
  log_message "Extracting the downloaded package..."
  if [ "$PLATFORM" == "Linux" ]; then
    $STAR --checkpoint=.1000 -jxf $TMP_DIR/$CEF_DOWNLOAD_FILE -C $TMP_DIR/ $CEF_RESOURCE_DIR
    $CEF_RELEASE_DIR
  else
    $UNZIP "$TMP_DIR/$CEF_DOWNLOAD_FILE" -d "$TMP_DIR/"
  fi
  if [ $? -ne 0 ]; then
    return_error 6 "Error: Extraction failed. Please try again"
  fi
  log_message "Extraction successful"
}
```

```

function copy_cef()
{
log_message "Copying the $CEF_DISPLAY_NAME files..."
if [ -d "$CEF_INSTALL_DIR" ]; then
    $RM -rf "$CEF_INSTALL_DIR/*"
else
    if [ "$PLATFORM" == "Linux" ]; then
        $MKDIR -p "$CEF_INSTALL_DIR"
    fi
fi

attempting_copy=true
if [ "$PLATFORM" == "Linux" ]; then
    $CP -r $TMP_DIR/$CEF_RESOURCE_DIR $CEF_INSTALL_DIR/
    if [ $? -ne 0 ]; then
        return_error 7 "Error: Copying of files to $CEF_INSTALL_DIR failed"
    fi

    $CP -r $TMP_DIR/$CEF_RELEASE_DIR $CEF_INSTALL_DIR/
    if [ $? -ne 0 ]; then
        return_error 7 "Error: Copying of files to $CEF_INSTALL_DIR failed"
    fi

    $CP -r $CEF_INSTALL_DIR/Resources/* $CEF_INSTALL_DIR/Release/
    if [ $? -ne 0 ]; then
        return_error 7 "Error: Copying of files to $CEF_INSTALL_DIR failed"
    fi
else
    $CP -rf "$TMP_DIR/${CEF_DIRNAME}" "${CEF_INSTALL_DIR2}"
    if [ $? -ne 0 ]; then
        return_error 7 "Error: Copying of files to $CEF_INSTALL_DIR failed"
    fi
fi
log_message "Copying successful"
}

```


function install()

```
{
log_message "$CEF_DISPLAY_NAME"
if [ ! -d "$TMP_DIR" ]; then
    $MKDIR -p "$TMP_DIR"
fi

if [ "$REINSTALL_CEF" == "false" ]; then
    check_installed
    if [ "$ALREADY_INSTALLED" == "true" ]; then
        return_error 0 "Not installing again. Please use 'reinstall' option for reinstallation."
    fi
fi

check_root_privilege
if [ ! -d "$PULSE_LOG_DIR" ]; then
    $MKDIR -p "$PULSE_LOG_DIR"
fi
LOG_DIR="$PULSE_LOG_DIR"
check_disk_space
download_cef
validate_checksum
extract_cef
copy_cef
cleanup
log_message "Successfully installed $CEF_DISPLAY_NAME"
}
```

function uninstall()

```
{
  check_root_privilege
  if [ -d "$CEF_INSTALL_DIR" ]; then
    $RM -rf "$CEF_INSTALL_DIR"
    if [ $? -ne 0 ]; then
      return_error 8 "Error: Removal of $CEF_INSTALL_DIR failed"
    else
      log_message "Successfully uninstalled $CEF_DISPLAY_NAME"
    fi
  fi
}
#
# Main
#

if [ ! -d "$LOG_DIR" ] || [ $EUID -ne 0 ]; then
  LOG_DIR=/tmp
fi

if [ "$1" == "check_installed" ]; then
  check_installed
elif [ "$1" == "install" ] || [ "$1" == "reinstall" ]; then
  if [ "$2" == "-$TMP_DIR_PARAM" ]; then
    if [ "$3" != "" ]; then
      TMP_DIR=$3
    fi
  elif [ "$2" != "" ]; then
    printUsage
  fi
  if [ "$1" == "reinstall" ]; then
    REINSTALL_CEF=true
  fi
  install
elif [ "$1" == "uninstall" ]; then
  uninstall
else
  printUsage
fi
```