Procedure to create Installers

*Rev: A0-03*

*18 jan, 2017*

Spectra Logic, Inc.

Deep Storage Browser

Procedure to create Installers for windows, macOS , and Ubuntu platforms.

**Revision History:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Rev No.** | **Description** | **By** |
| 01-AUG-2016 | A0-01 | First draft | VVDN |
| 19-SEP-2016 | A0-02 | Second draft | VVDN |
| 11-NOV-2016 | A0-03 | Third draft | VVDN |

Table of Contents

[1 Steps to create a EXE file from runnable jar for Windows 4](#_Toc467064873)

[1.1 Build the project 4](#_Toc467064874)

[1.2 Install WiX Toolset and visual studio for windows 4](#_Toc467064875)

[1.3 Create a project in visual studio 4](#_Toc467064876)

[1.4 Create a custom action. 6](#_Toc467064877)

[1.5 Pack required java with the project 7](#_Toc467064878)

[1.6 Place the runnable jar and installer icon 7](#_Toc467064879)

[1.7 Create run.bat file and add it to the project 7](#_Toc467064880)

[1.8 Build the .msi 10](#_Toc467064881)

[2 Steps to create a Debian package from an runnable jar for Ubuntu 11](#_Toc467064882)

[2.1 Create a directory where .deb file will be stored. 11](#_Toc467064883)

[2.2 Copy jar and icon 11](#_Toc467064884)

[2.3 Create a script to start the java application. 11](#_Toc467064885)

[2.4 Create a desktop file which contains categorized launcher for the application. 11](#_Toc467064886)

[2.5 Create a control file, containing metadata about the package, including (most importantly) its dependencies. 12](#_Toc467064887)

[2.6 Debian requires a copyright notice, so create a copyright file 12](#_Toc467064888)

[2.7 Create preinst file to perform actions before installation of software on linux 12](#_Toc467064889)

[2.8 Create postrm file to remove extra files and folder on uninstallation of the software 14](#_Toc467064890)

[2.9 Finally write a shell script to wrap it all up and deliver a .deb package 14](#_Toc467064891)

[2.10 Run the script as root: 16](#_Toc467064892)

[2.11 Run lintian on the result and make adjustments where necessary. 16](#_Toc467064893)

[2.12 run .deb package by double click 16](#_Toc467064894)

[3 Steps to create .dmg file from runnable jar for MAC 16](#_Toc467064895)

[3.1 create a folder where you want to create .dmg file 16](#_Toc467064896)

[3.2 create icon for the application 17](#_Toc467064897)

[3.3 Create dmg file 17](#_Toc467064898)

# Steps to create a EXE file from runnable jar for Windows

## Build the project

* Build the project to create runnable jar say "**dsb.jar**"

## Install WiX Toolset and visual studio for windows

* Install Visual Studio 2015 along with latest WiX toolset.

## Create a project in visual studio

* Create a windows installer project say "**DSBSetupProject**" in visual studio.
* A default file named "Product.wxs" will be created. Add the following code in that file to make the UI of the installer for windows.

**Product.wxs**

<?xmlversion="1.0"encoding="UTF-8"?>

<!-- The name of the product -->

<?defineName = "Deep Storage Browser" ?>

<!-- The manufacturer, for setup package publisher and folder info -->

<?defineManufacturer = "Spectra Logic" ?>

<!-- The version -->

<?defineVersion = "2.0.0" ?>

<Wixxmlns="http://schemas.microsoft.com/wix/2006/wi">

<?defineTestDSB2\_TargetDir=$(var.TestDSB2.TargetDir)?>

<ProductId="79aa821f-d3ac-422b-a9a8-a4b8a0a6f927"Name="$(var.Name)"Language="1033"Version="$(var.Version)"Manufacturer="$(var.Manufacturer)"UpgradeCode="61718ba5-b4c2-43b8-838b-fb3930c5ed3f">

<PackageInstallerVersion="200"Compressed="yes"InstallScope="perMachine"Platform="x64"/>

<IconId="icon.ico"SourceFile="$(var.ProjectDir)deep\_storage\_browser.ico" />

<PropertyId="ARPPRODUCTICON"Value="icon.ico" />

<UI>

<PublishDialog="WelcomeDlg"

Control="Next"

Event="NewDialog"

Value="InstallDirDlg"

Order="5">1</Publish>

<PublishDialog="InstallDirDlg"

Control="Back"

Event="NewDialog"

Value="WelcomeDlg"

Order="5">1</Publish>

</UI>

<PropertyId="WIXUI\_INSTALLDIR"Value="INSTALLFOLDER" />

<UIRefId="WixUI\_InstallDir" />

<MajorUpgradeDowngradeErrorMessage="A newer version of [ProductName] is already installed." />

<MediaTemplateEmbedCab="yes"/>

<BinaryId="CustomAction.CA.dll"SourceFile="$(var.ProjectDir)CustomAction.CA.dll" />

<InstallExecuteSequence>

<Custom Action="Uninstall" Before="InstallFinalize" >(NOT UPGRADINGPRODUCTCODE) AND (REMOVE="ALL")</Custom>

</InstallExecuteSequence>

<CustomActionId="DeleteLogFile"Return="check"Execute="immediate"

BinaryKey="CustomAction.CA.dll"DllEntry="DeleteLogFile"/>

<CustomActionId="SetLaunchApplicationTarget"Property="WixShellExecTarget"Value="[#run.bat]" />

<CustomActionId="LaunchApplication"BinaryKey="WixCA"DllEntry="WixShellExec"Impersonate="yes" />

<UIId="LaunchApplication">

<PublishDialog="ExitDialog"Control="Finish"Event="DoAction"Value="SetLaunchApplicationTarget">

<![CDATA[NOT Installed]]>

</Publish>

<PublishDialog="ExitDialog"Control="Finish"Event="DoAction"Value="LaunchApplication">

<![CDATA[NOT Installed]]>

</Publish>

</UI>

<FeatureId="ProductFeature"Title="$(var.Name)"Level="1">

<ComponentRefId="ApplicationShortcut" />

<ComponentRefId="ApplicationShortcutDesktop" />

<ComponentRefId="dsb.jar" />

<ComponentRefId="run.bat" />

<ComponentGroupRefId="JavaFilesGroup"/>

</Feature>

</Product>

<Fragment>

<DirectoryRefId="ApplicationProgramsFolder">

<ComponentId="ApplicationShortcut"Guid="{B48BD021-AD51-4811-A5B9-A70892BD7334}">

<ShortcutId="ApplicationStartMenuShortcut"Name="$(var.Name)"Icon="icon.ico"Target="[#run.bat]"WorkingDirectory="INSTALLFOLDER" />

<RemoveFolderId="ApplicationProgramsFolder"On="both" />

<RegistryValueRoot="HKCU"Key="Software\Spectra Logic\$(var.Name)"Name="ApplicationShortcut"Type="integer"Value="1"KeyPath="yes" />

</Component>

</DirectoryRef>

<DirectoryRefId="DesktopFolder">

<ComponentId="ApplicationShortcutDesktop"Guid="{8213EAF3-AFB1-4946-9A79-99A8543732E7}">

<ShortcutId="ApplicationDesktopShortcut"Name="$(var.Name)"Icon="icon.ico"Target="[#run.bat]"WorkingDirectory="INSTALLFOLDER" />

<RemoveFolderId="DesktopFolder"On="both" />

<RegistryValueRoot="HKCU"Key="Software\Spectra Logic\$(var.Name)"Name="ApplicationDesktopShortcut"Type="integer"Value="1"KeyPath="yes" />

</Component>

</DirectoryRef>

<DirectoryId="TARGETDIR"Name="SourceDir">

<DirectoryId="ProgramFiles64Folder">

<DirectoryId="ROOTDIRECTORY"Name="$(var.Manufacturer)">

<DirectoryId="INSTALLFOLDER"Name="$(var.Name)">

<ComponentId="dsb.jar"Guid="{AFF58ABB-1ED8-4860-8AE1-38DDC3C45B60}"Win64="yes">

<FileId="dsb.jar"Source="dsb.jar"KeyPath="yes" />

</Component>

<ComponentId="run.bat"Guid="{fabf23ea-a1b3-4387-ac5e-e560e934604a}"Win64="yes">

<FileId="run.bat"Source="run.bat"KeyPath="yes" />

</Component>

</Directory>

</Directory>

</Directory>

<DirectoryId="ProgramMenuFolder">

<DirectoryId="ApplicationProgramsFolder"Name="Spectra Logic" />

</Directory>

<DirectoryId="DesktopFolder"Name="desktop">

</Directory>

</Directory>

</Fragment>

</Wix>

## Create a custom action.

* Create a custom action in DSBSetupProject to add extra functionality to the installer like deletion of extra files or folders while unstallation, which are outside the installer folder.

**CustomAction.cs**

using System;

usingSystem.Collections.Generic;

usingSystem.Text;

usingMicrosoft.Deployment.WindowsInstaller;

using System.IO;

usingSystem.Windows;

namespaceCustomAction

{

publicclassCustomActions

{

[CustomAction]

publicstaticActionResult CustomAction1(Session session)

{

session.Log("Begin CustomAction1");

returnActionResult.Success;

}

[CustomAction]

publicstaticActionResultDeleteLogFile(Session session)

{

try

{

varfileName = Environment.GetFolderPath(

Environment.SpecialFolder.UserProfile);

string path = System.IO.Path.Combine(fileName + "\\.dsbrowser");

if (Directory.Exists(path))

{

Directory.Delete(path, true);

}

}

catch (Exception ex)

{

session.Log("Inside DeleteLogFile" + ex);

}

returnActionResult.Success;

}

**Pass its reference to the Product.wxs file.**

## Pack required java with the project

* Create a folder named "java" within the project directory (i.e. inside DSBSetupProject folder)
* Copy required jre (for this project jre1.8.0\_102) folder containing all the jars and libraries, inside this java folder.
* Create a .bat file in the same directory (inside DSBSetupProject folder) say createWXSJavaFiles.bat, that must contain the following lin of code:

"%WIX%\bin\heat.exe" dir "Java" -ag -sfrag -dr "INSTALLFOLDER" -cg JavaFilesGroup -varvar.JavaPath -out javaFiles.wxspowershell -Command "(gcjavaFiles.wxs) -replace '<Component ', '<Component Win64=\"yes\" ' | Out-File javaFiles.wxs" -encoding UTF8

**This will create JavaFiles.wxs file when we will run it via cmd with in the same directory where this .bat file is situated, this is responsible for binding required java for our project.**

## Place the runnable jar and installer icon

* Copy the runnable jar i.e. dsb.jar file inside the DSBSetupProject folder.
* Copy installer icon (dsb.ico) in the same directory.

**Pass their references to the Product.wxs file.**

## Create run.bat file and add it to the project

* Create run.bat file to run the dsb.jar file using packed java libraries. copy it to DSBSetupProject Folder.

**run.bat**

if "%DEBUG%" == "" @echo off

@rem ##########################################################################

@rem

@rem DeepStorageBrowser startup script for Windows

@rem

@rem ##########################################################################

@rem Set local scope for the variables with windows NT shell

if "%OS%"=="Windows\_NT" setlocal

set DIRNAME=%~dp0

if "%DIRNAME%" == "" set DIRNAME=.

set APP\_BASE\_NAME=%~n0

set APP\_HOME=%DIRNAME%

@rem Add default JVM options here. You can also use JAVA\_OPTS and NCL\_GUI\_OPTS to pass JVM options to this script.

set DEFAULT\_JVM\_OPTS=

set NCL\_GUI\_OPTS=

@rem Find java.exe

set JAVA\_HOME=Java\jre1.8.0\_102

if defined JAVA\_HOME gotofindJavaFromJavaHome

set JAVA\_EXE=javaw.exe

%JAVA\_EXE% -version >NUL 2>&1

if "%ERRORLEVEL%" == "0" gotoinit

echo.

echo ERROR: JAVA\_HOME is not set and no 'java' command could be found in your PATH.

echo.

echo Please set the JAVA\_HOME variable in your environment to match the

echo location of your Java installation.

goto fail

:findJavaFromJavaHome

set JAVA\_HOME=%JAVA\_HOME:"=%

set JAVA\_EXE=%JAVA\_HOME%/bin/javaw.exe

if exist "%JAVA\_EXE%" gotoinit

echo.

echo ERROR: JAVA\_HOME is set to an invalid directory: %JAVA\_HOME%

echo.

echo Please set the JAVA\_HOME variable in your environment to match the

echo location of your Java installation.

goto fail

:init

@rem Get command-line arguments, handling Windows variants

if not "%OS%" == "Windows\_NT" goto win9xME\_args

if "%@eval[2+2]" == "4" goto 4NT\_args

:win9xME\_args

@rem Slurp the command line arguments.

set CMD\_LINE\_ARGS=

set \_SKIP=2

:win9xME\_args\_slurp

if "x%~1" == "x" goto execute

set CMD\_LINE\_ARGS=%\*

goto execute

:4NT\_args

@rem Get arguments from the 4NT Shell from JP Software

set CMD\_LINE\_ARGS=%$

:execute

@rem Setup the command line

@rem Execute ncl-gui

START %JAVA\_EXE% -jar dsb.jar

:end

@rem End local scope for the variables with windows NT shell

if "%ERRORLEVEL%"=="0" gotomainEnd

:fail

rem Set variable NCL\_GUI\_EXIT\_CONSOLE if you need the \_script\_ return code instead of

rem the \_cmd.exe /c\_ return code!

if not "" == "%NCL\_GUI\_EXIT\_CONSOLE%" exit 1

exit /b 1

:mainEnd

if "%OS%"=="Windows\_NT" endlocal

:omega

## Build the .msi

* Build the project by right click on DSBSetupProject, this will create .msi (DeepStorageBrowser.msi) file with in the bin->debug folder.
* Double click on that we are ready to install the software.

# Steps to create a Debian package from an runnable jar for Ubuntu

## Create a directory where .deb file will be stored.

->mkdir spectralogic

->cd spectralogic

## Copy jar and icon

* Copy the required jar i.e spectralogic.jar and icon of the installer.

## Create a script to start the java application.

**deepstoragebrowser**

#!/bin/sh

/bin/ps -ef | grep -i spectralogic.jar | grep -v "grep" >/dev/null 2>&1

if [ $? -eq0 ]

then

echo "Already running..!!"

wmctrl -a Deep Storage Browser

else

java -jar /usr/share/deepstoragebrowser/spectralogic.jar

fi

## Create a desktop file which contains categorized launcher for the application.

**deepstoragebrowser.desktop**

[Desktop Entry]

Encoding=UTF-8

Name=Deep Storage Browser

Comment=DeepStorageBrowser

Exec=/usr/deepstoragebrowser/deepstoragebrowser

Icon=/usr/share/deepstoragebrowser/deep\_storage\_browser.png

Terminal=false

Type=Application

StartupNotify=true

Name[en\_IN]=deepstoragebrowser

## Create a control file, containing metadata about the package, including (most importantly) its dependencies.

**control**

Section: java

Priority: optional

Architecture: all

Maintainer: VVDN Technologies

Description: Deep Storage Browser.

\*The Deep Storage Browser is a free and open-source BlackPearl Spectra S3 client that provides a graphical FTP-like interface to manage files between BlackPearl and your local computer or a server. Spectra S3 enables you to efficiently move large quantities of data with minimal cost and effort. The Spectra S3 communication interface allows the client to direct and manage “bulk” storage read or write operations of data objects. This interface supports bulk object storage operations with tape for accessibility to the lowest cost media option.

Depends: openjdk-8-jre | openjdk-7-jre | openjdk-6-jre | oracle-java8-installer | oracle-java7-installer | oracle-java6-installer | jarwrapper

## Debian requires a copyright notice, so create a copyright file

**copyright**

This package was debianized by VVDN technologies

Monday July 26 2016.

Copyright: (c) 2016 VVDN Technologies

## Create preinst file to perform actions before installation of software on linux

**preinst**

#!/bin/sh

set -x

a=`dpkg-query -l | grepinotify-tools | wc -l`

if [ $a -eq 0 ]

then

sudo apt-get install inotify-tools

else

echo "Already Installed"

notify-send "inotify-tools" "Already Installed"

fi

sudomkdir -p /opt/jvm

cd /opt

if [ -f "/opt/jdk-8u102-linux-i586.tar.gz" ]

then

if [ -d "/opt/jdk1.8.0\_102" ]

then

echo "jdk directory found"

notify-send "JDK" "jdk directory found"

else

notify-send "JDK-tar.gz is present in the system"

sudo tar -xvzf "jdk-8u102-linux-i586.tar.gz"

fi

else

sudowget --no-cookies --no-check-certificate --header "Cookie: gpw\_le24=http%3A%2F%2Fwww.oracle.com%2F; oraclelicense=accept-securebackup-cookie" "http://download.oracle.com/otn-pub/java/jdk/8u102-b14/jdk-8u102-linux-i586.tar.gz"

command&& notify-send 'JDK' 'JDK-tar.gz has downloaded successfully'

sudo tar -xvzf jdk-8u102-linux-x64.tar.gz

echo "Already installed"

notify-send "JDK" "jdk Already installed"

fi

sudo update-alternatives --install /usr/bin/java java /opt/jdk1.8.0\_102/bin/java 1100

a=`dpkg-query -l | grepwmctrl | wc -l`

if [ $a -eq 0 ]

then

sudo apt-get install wmctrl

else

echo "Already installed"

notify-send "wmctrl" "Already Installed"

fi

exit 0

## Create postrm file to remove extra files and folder on uninstallation of the software

**postrm**

#!/bin/sh

set -x

echo "postrm executed"

MY\_DIR=$HOME"/.dsbrowser"

if [ -d $MY\_DIR ]; then

rm -rf .??\* $MY\_DIR

fi

notify-send "Removed .dsbrowser folder"

exit 0

## Finally write a shell script to wrap it all up and deliver a .deb package

**packagedeb.sh**

#!/bin/sh

PACKAGE\_NAME="deepstoragebrowser"

PACKAGE\_VERSION="2.0"

SOURCE\_DIR=$PWD

TEMP\_DIR="/tmp"

mkdir -p $TEMP\_DIR/debian/DEBIAN

mkdir -p $TEMP\_DIR/debian/lib

mkdir -p $TEMP\_DIR/debian/usr/uninstall

mkdir -p $TEMP\_DIR/debian/usr/deepstoragebrowser

mkdir -p $TEMP\_DIR/debian/usr/share/applications

mkdir -p $TEMP\_DIR/debian/usr/share/$PACKAGE\_NAME

mkdir -p $TEMP\_DIR/debian/usr/share/doc/$PACKAGE\_NAME

mkdir -p $TEMP\_DIR/debian/usr/share/common-licenses/$PACKAGE\_NAME

sh uninstall

echo "hello"

echo "Package: $PACKAGE\_NAME" > $TEMP\_DIR/debian/DEBIAN/control

echo "Version: $PACKAGE\_VERSION" >> $TEMP\_DIR/debian/DEBIAN/control

cat control >> $TEMP\_DIR/debian/DEBIAN/control

catpostrm>> $TEMP\_DIR/debian/DEBIAN/postrm

chmod 0755 $TEMP\_DIR/debian/DEBIAN/postrm

catpreinst>> $TEMP\_DIR/debian/DEBIAN/preinst

chmod 0755 $TEMP\_DIR/debian/DEBIAN/preinst

cp \*.desktop $TEMP\_DIR/debian/usr/share/applications/

cp copyright $TEMP\_DIR/debian/usr/share/common-licenses/$PACKAGE\_NAME/ # results in no copyright warning

cp copyright $TEMP\_DIR/debian/usr/share/doc/$PACKAGE\_NAME/ # results in obsolete location warning

echohii 11

cp \*.jar $TEMP\_DIR/debian/usr/share/$PACKAGE\_NAME/

cp uninstall $TEMP\_DIR/debian/usr/uninstall/

echohii 22

cp $PACKAGE\_NAME $TEMP\_DIR/debian/usr/deepstoragebrowser/

echo "$PACKAGE\_NAME ($PACKAGE\_VERSION) trusty; urgency=low" >changelog

echo " \* Rebuild" >>changelog

echo " -- Spectra Logic Corporation <www.spectralogic.com> `date -R`" >>changelog

gzip -9c changelog> $TEMP\_DIR/debian/usr/share/doc/$PACKAGE\_NAME/changelog.gz

echopng

cp \*.png $TEMP\_DIR/debian/usr/share/$PACKAGE\_NAME/

chmod 0664 $TEMP\_DIR/debian/usr/share/$PACKAGE\_NAME/\*png

echopng done

PACKAGE\_SIZE=`du -bs $TEMP\_DIR/debian | cut -f 1`

PACKAGE\_SIZE=$((PACKAGE\_SIZE/1024))

echo "Installed-Size: $PACKAGE\_SIZE" >> $TEMP\_DIR/debian/DEBIAN/control

chown -R root $TEMP\_DIR/debian/

chgrp -R root $TEMP\_DIR/debian/

cd $TEMP\_DIR/

dpkg --build debian

mvdebian.deb $SOURCE\_DIR/$PACKAGE\_NAME-$PACKAGE\_VERSION.deb

rm -r $TEMP\_DIR/debian

## Run the script as root:

Need to change modes of some files

chmod 0644 "jar name" like spectralogic.jar

chmod +x "name of file that contains jar's path" like deepstoragebrowser

chmod 777 deepstoragebrowser.desktop

**sudo** .**/**packagedeb.sh

## Run lintian on the result and make adjustments where necessary.

lintian-i spectralogic-0.1.1.deb

## run .deb package by double click

* .deb package (deepstoragebrowser\_2.0.deb) is ready to be installed on the system.

# Steps to create .dmg file from runnable jar for MAC

## create a folder where you want to create .dmg file

->mkdirDeepStorageBrowser

->cdDeepStorageBrowser

## create icon for the application

->curl"image path" say "https://github.com/SpectraLogic/ds3\_java\_browser/blob/master/Builders/Mac/DeepStorageBrowser.png". Alternatively, the file can also be downloaded directly form github to your local path to perform the steps mentioned below.

->mkdir"foldername" say "DeepStorageBrowser.iconset"

->sips-z 32 32DeepStorageBrowser.png --out DeepStorageBrowser.iconset/icon\_32x32.png//giving dimensions for the icon

->sips-s format icns DeepStorageBrowser.png --out DeepStorageBrowser.icns//converts the format of icon from png to icns

->ls-l

->mkdir-p package/macosx//make directory for icon

->cpDeepStorageBrowser.icns package/macosx//copy icon to that directory

## Createdmg file

->jdk=$(/usr/libexec/java\_home)//java\_home reference in jdk variable

->$jdk/bin/javapackager -version//to check the version of present java

->By using AppMaker we madeDeepStorageBrowser.app//Appmaker used to attach .jar and .icns to make .app

->$jdk/bin/javapackager -deploy -native dmg -srcfiles DeepStorageBrowser.app -appclasscom.spectralogic.dsbrowser.gui.Main -name DeepStorageBrowser -BappVersion=2.1 -Bmac.signing-key-developer-id-app="Spectra Logic Corporation" -outdir deploy -outfileDeepStorageBrowser// creates dmg file in deploy folder

Some Important Points while creating .dmg file:

* For version name , just type the –BappVersion=2.0 after dmg file name as:

$jdk/bin/javapackager -deploy -native dmg -srcfiles "SpectraLogic.jar" -appclass "com.spectralogic.dsbrowser.gui.Main" -name "spectraLogic"–BappVersion=2.0-outdirdeploy -outfilespectraLogic –v

* Dmg file name and –outfile name should be same.
* Image (.icns) and folder name should be same.
* DeepStorageBrowser.app should be in same folder where all other files are placed.
* Keep .jar file in the same folder as well. (in DeepStorageBrowser)