

# Running Camry Impact Model in LS-DYNA<sup>®</sup> format using **OpenRadioss** and **Apptainer**

[Altair Radioss is now available as an open source project!](#) This is major news for the simulation community with many contributors already onboard with [OpenRadioss](#). Moving into open source can speed collaborative efforts like containerizing the application and testing on a new operating system. That's exactly what we at CIQ did within days of Altair's announcement.

This article describes two things, (1) How to create OpenRadioss **Apptainer** container image from daily released repository and (2) How to run 2012 Toyota Camry Detailed Finite Element Model [Toyota Camry Impact Model in LS-DYNA<sup>®</sup> format](#) using OpenRadioss Apptainer container. We tested this procedures on **Rocky Linux 8.6**.

## Installation

The steps below assume that Rocky Linux 8 is already installed as the operating system. This post does not cover the steps to install the OS.

### Apptainer

The first step is to install Apptainer on Rocky Linux 8.

```
1 $ wget -O /tmp/apptainer-1.0.3-1.x86_64.rpm https://github.com/apptainer/apptainer/releases/download/v1.0.3/apptainer-1.0.3-1.x86_64.rpm
2 $ sudo dnf install -y /tmp/apptainer-1.0.3-1.x86_64.rpm
```

### Definition File

The steps below pull in the necessary tools and libraries, grabs the OpenRadioss [latest-20220920](#) release from **source code** that is available on OpenRadioss GitHub repository, then builds it.

These steps go into a definition file that is then used to build the container

Create the definition file called [openradioss-20220920.def](#) and add (or cut & paste) the information shown below.

```
1 Bootstrap: docker
2 From: rockylinux:8.6
3
4 %post
5 dnf install -y \
6 gcc gcc-gfortran gcc-c++ make cmake perl git-lfs \
7 wget git patch
8 cd /tmp
9 wget https://download.open-mpi.org/release/open-mpi/v4.1/openmpi-4.1.2.tar.gz
10 tar xzvf openmpi-4.1.2.tar.gz
11 cd openmpi-4.1.2
12 ./configure --prefix=/opt/openmpi
13 make -j
14 make install
15 cd /opt
16 git lfs install
17 git clone --depth 1 --branch latest-20220920 https://github.com/OpenRadioss/OpenRadioss.git
18 cd /opt/OpenRadioss/starter
```

```
19 ./build_script.sh -arch=linux64_gf
20 cd /opt/OpenRadioss/engine
21 ./build_script.sh -arch=linux64_gf -mpi=mpi
22 cd /opt/OpenRadioss/tools/anim_to_vtk/linux64
23 ./build.bash
24 cd /opt/OpenRadioss/tools/th_to_csv/linux64
25 ./build.bash
26
27 %environment
28 export OPENRADIOSS_PATH=/opt/OpenRadioss
29 export RAD_CFG_PATH=$OPENRADIOSS_PATH/hm_cfg_files
30 export OMP_STACKSIZE=400m
31
32 export PATH=$OPENRADIOSS_PATH/exec:$PATH
33 export LD_LIBRARY_PATH=$OPENRADIOSS_PATH/extlib/hm_reader/linux64/:$OPENRADIOSS_PATH/extlib/h3d/lib/linux64/:$LD
34
35 export PATH=/opt/openmpi/bin:$PATH
36 export LD_LIBRARY_PATH=/opt/openmpi/lib:$LD_LIBRARY_PATH
```

### Build OpenRadioss Apptainer Image (SIF format)

Now we can build the OpenRadioss [latest-20220920](#) Apptainer image with the following command:

```
1 $ sudo apptainer build openradioss-20220920.sif openradioss-20220920.def
```

Copy this Apptainer image to somewhere in your PATH

```
1 $ sudo cp openradioss-20220920.sif /usr/local/bin
```

Containerized! We're ready to utilize OpenRadioss using this [openradioss-20220919.sif](#) Apptainer on your compute resources. This can run on your laptop, workstation, on-prem HPC cluster and cloud resources.

### MPI

Before running a model, we do need to install OpenMPI on the system if it's not already installed.

```
1 $ sudo dnf install -y openmpi numactl hwloc
```

Close your terminal, then open the terminal again to apply the changes. Now you can load OpenMPI module using the `module` command.

## Simulation Setup

Now that we have the container, we want to test it with the Toyota Camry workload. This is a crash simulation, and we will use ParaView to visualize the result.

### Download 2012 Toyota Camry Impact Model in LS-DYNA® format

Download the model from [CCSA](#), and download the [OpenRadioss launch file](#) for this file.

```
1 $ wget https://media.ccsa.gmu.edu/model/2012-toyota-camry-detailed-v5a.zip
2 $ unzip 2012-toyota-camry-detailed-v5a.zip
3 $ wget -O CamryOpenRadioss.key https://openradioss.atlassian.net/wiki/download/attachments/10518559/CamryOpenRadi
4 $ mv CamryOpenRadioss.key 2012-toyota-camry-detailed-v5a
5 $ cd 2012-toyota-camry-detailed-v5a
6 # Download attached based_on_paully_workaround.patch and apply
7 $ patch -p1 < based_on_paully_workaround.patch
```

## Running OpenRadioss using MPI

Go to the model directory and then load OpenMPI module.

```
1 $ module load mpi/openmpi-x86_64
```

Run **starter** first, and then run **engine** using OpenMPI. Here we specify 8 MPI processes to run this simulation

```
1 $ openradioss-20220920.sif starter_linux64_gf -i CamryOpenRadioss.key -np 8
2 $ mpirun -np 8 openradioss-20220920.sif engine_linux64_gf_ompi -i CamryOpenRadioss_0001.rad
```

## Output

If it runs successfully, it will shows you **NORMAL TERMINATION** .

```
1 ELAPSED TIME = 80054.69 s
2 22:14:14
3 ESTIMATED SPEEDUP= 7.95
4
5 NORMAL TERMINATION
6 TOTAL NUMBER OF CYCLES : 172244
```

## Post-processing using ParaView

Now we want to see our results using ParaView. The Toyota Camry simulation output files are in ANIM format. We need to convert them to VTK format and install ParaView. The we can see our results.

### Convert output files ANIM to VTK format

To visualize ANIM format with ParaView, we need to **convert ANIM format to VTK format**. The **latest-20220913** OpenRadioss includes a converter utility **anim\_to\_vtk** inside tools directory.

A directory listing shows we have 25 output files from our simulation to convert, and each file has a three digit number, from 001 to 025, at the end of the name.

```
1 $ ls CamryOpenRadiossA*
2 CamryOpenRadiossA001 CamryOpenRadiossA005 CamryOpenRadiossA009 CamryOpenRadiosA013 CamryOpenRadiossA017 Camr
3 CamryOpenRadiossA002 CamryOpenRadiossA006 CamryOpenRadiossA010 CamryOpenRadiosA014 CamryOpenRadiossA018 Camr
4 CamryOpenRadiossA003 CamryOpenRadiossA007 CamryOpenRadiossA011 CamryOpenRadiosA015 CamryOpenRadiossA019 Camr
5 CamryOpenRadiossA004 CamryOpenRadiossA008 CamryOpenRadiossA012 CamryOpenRadiosA016 CamryOpenRadiossA020 Camr
```

The following command will convert all 25 files from ANIM format to VTK format.

```
1 $ seq -f CamryOpenRadiossA%03g 025 | xargs -L 1 -I{} sh -c 'openradioss-20220920.sif anim_to_vtk.linux64.exe "$1"
```

Now our directory listing shows our simulation results in VTK format ready to rendering by ParaView.

```
1 $ ls CamryOpenRadiossA*.vtk
2 CamryOpenRadiossA001.vtk CamryOpenRadiossA006.vtk CamryOpenRadiossA011.vtk CamryOpenRadiossA016.vtk CamryOpen
3 CamryOpenRadiossA002.vtk CamryOpenRadiossA007.vtk CamryOpenRadiossA012.vtk CamryOpenRadiossA017.vtk CamryOpen
4 CamryOpenRadiossA003.vtk CamryOpenRadiossA008.vtk CamryOpenRadiossA013.vtk CamryOpenRadiossA018.vtk CamryOpen
5 CamryOpenRadiossA004.vtk CamryOpenRadiossA009.vtk CamryOpenRadiossA014.vtk CamryOpenRadiossA019.vtk CamryOpen
6 CamryOpenRadiossA005.vtk CamryOpenRadiossA010.vtk CamryOpenRadiossA015.vtk CamryOpenRadiossA020.vtk CamryOpen
```

## Install ParaView

Download ParaView from their [official website](https://www.paraview.org/) and place it to your preferred location.

```
1 $ wget https://www.paraview.org/paraview-downloads/download.php?submit=Download&version=v5.10&type=binary&os=Linu
2 $ tar zxvf ParaView-5.10.1-MPI-Linux-Python3.9-x86_64.tar.gz
3 $ sudo mkdir -p /opt/ParaView
4 $ sudo mv ParaView-5.10.1-MPI-Linux-Python3.9-x86_64 /opt/ParaView/5.10.1
5 $ rm -rf ParaView-5.10.1-MPI-Linux-Python3.9-x86_64.tar.gz
6 $ export PATH=/opt/ParaView/5.10.1/bin:$PATH
7 $ export LD_LIBRARY_PATH=/opt/ParaView/5.10.1/lib:$LD_LIBRARY_PATH
```

## Rendering the results

To visualize our results, simply call ParaView providing it which file to render. Here is the command and resulting visualization for `CamryOpenRadiossA020.vtk`

```
1 $ paraview --data=CamryOpenRadiossA020.vtk
```

The pictures below are shown stress magnitude. The both images are rendering of `CamryOpenRadiossA020.vtk`. The result shows most of the stress absorbed at front end and much less at passenger cabin.

