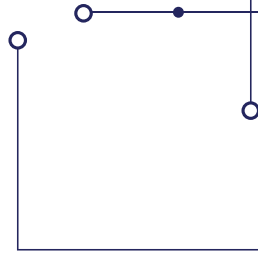




How to convert your LoadRunner script to Gatling



Purpose of this e-guide

After a brief presentation of Gatling and LoadRunner, we will learn a way to migrate your existing LoadRunner script into a Gatling simulation in just a few minutes!

Presentation of the tools

Gatling is an Open-Source load testing tool (<https://github.com/gatling/gatling>), created by the company Gatling Corp. Gatling Corp also develops an enterprise solution over Gatling: Gatling Enterprise, which has advanced features with the same core engine.

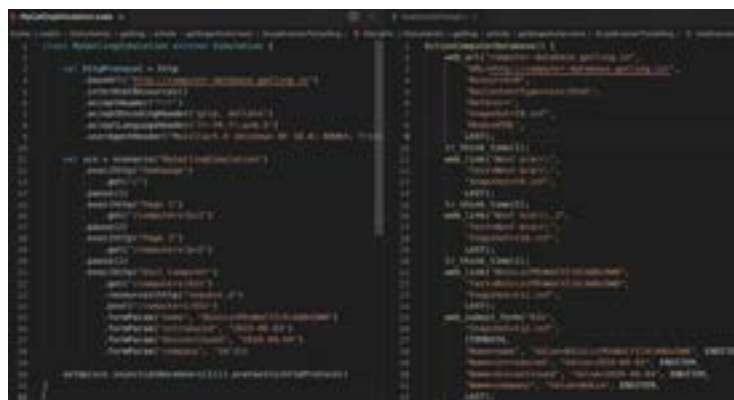
This e-guide will be dedicated to the Open-Source version.

LoadRunner is a load testing tool from Micro Focus. For this e-guide, we'll use the community edition of LoadRunner (limited to 50 users), available [here](#).

Let's quickly compare the features of both tools:

Criteria	Gatling	LoadRunner
Platform	Java application, can be run with a build tool	Windows application
Licensing	Free (Open-Source), enterprise version available C	Community edition available for up to 50 virtual users, complete pricing here
Coding Language	Scala (specific DSL)	C / Javascript for the HTTP protocol
Protocol support	HTTP, JMS, MQTT, Kafka	Various protocols including HTTP, SAP, LDAP, MQTT
Record and replay	Available	Available

You can see below the scripts of the same scenario, Gatling is on the left and LoadRunner on the right. In Gatling, the injection profile (arrival rate of the users) is also described in the script, while it is in the Runtime Settings for LoadRunner (not in the screenshot).



I Convert to Gatling

The easiest way to convert your LoadRunner script to a Gatling simulation is to replay your script with LoadRunner and record its result with the Gatling recorder.

I'll assume LoadRunner is already installed on your computer, with a working script. With this method, you'll be able to convert your "Web - HTTP/HTML" and "Web Services" scripts.

Step 1: Install Java

Gatling requires Java to run on your machine (minimum version we require is Java 8). If it is not installed, you can download it [here](#) by choosing the correct version according to your Operating System.

Step 2: Download Gatling

There are multiple ways to run Gatling: with the bundle, or with a build tool we support (Maven, Sbt, Gradle). We'll be working with the bundled version of Gatling which can be downloaded [here](#).

Step 3: Setup Gatling's recorder

Once you've downloaded the zip on Gatling website, extract it on your workspace and open the bin folder. In this folder, if you're using Windows, you'll find the gatling.bat file to launch the injection, and the recorder.bat file to create a Gatling simulation from your actions.

Double-click on recorder.bat to open the recorder configuration. If you want to change the simulation name, modify the **Class Name** input. If you want to change the port of the recorder proxy, modify Listening port (default is 8000). Click on start at the bottom-right to begin the recording.

Step 4: Setup LoadRunner

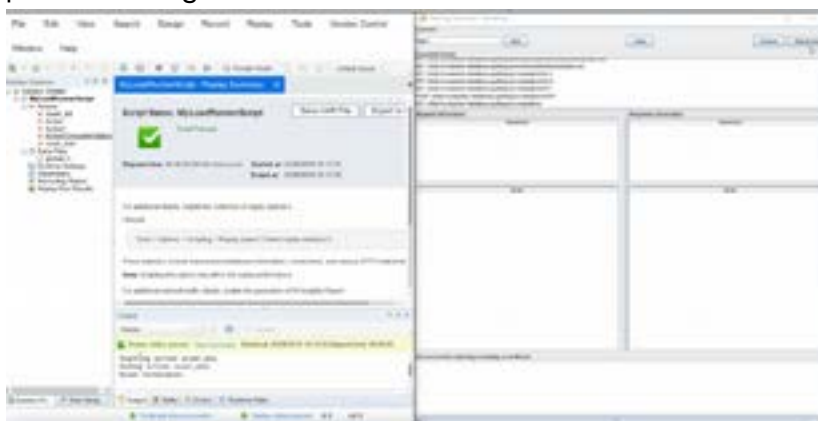
On your LoadRunner script, click on Runtime Settings and then Proxy. This is where we will configure LoadRunner to use Gatling proxy.

Choose Use custom proxy, then check Use proxy server. On HTTP proxy, fill localhost in the first field, and the recorder listening port in the second (default is 8000). You should also set the number of iterations as 1 in Run Logic and the think time as recorded in Think Time.

Step 5: Convert your script

Once the configuration is done, click on **Replay** to begin the injection with LoadRunner. If you monitor Gatling recorder, you will see every request is intercepted.

When the injection is finished, click on Stop & Save in the recorder. **Congratulations**, you just converted your LoadRunner script to a Gatling simulation!



| Launch your Gatling simulation

Your Gatling simulation is now available under:

```
{gatling-bundled}/user-files/simulations/RecordedSimulation.scala
```

This file is just a text file, so if you want to view/edit it, you can open it with a basic text editor (Notepad, Visual

Studio Code), or an IDE specialized in Scala (IntelliJ).

To run this simulation, double-click on the following file:

```
{gatling-bundled}/bin/gatling.bat
```

This will open a terminal, and launch Gatling inside it. You'll have to enter the number corresponding to your simulation (default is 0) and press enter. The run description is optional, so just press enter to start the injection.

Gatling will display the number of requests sent every 5 seconds, with the response time distribution at the end of the test. The link to the HTML report is displayed at the end of the test.



| Improve your Gatling Simulation

The simulation we created is basic, so you may want to improve it and add more things to it. Check the documentation for more information!

The first thing you should do is modify your injection profile by increasing your user arrival rate. If you want 1000

users to arrive on your application over a period of a minute, you may do:

```
setUp(scen.inject(rampUsers(1000) during (1 minute))).protocols(httpProtocol)
```

There is no limit to the concurrent number of users in Gatling, so don't hesitate to experiment! You should also rename your requests by logical names instead of the default request_n.

If you want to go further, I suggest you take a look at the advanced tutorial.

| Get to the next step

If you have more questions on how to migrate to Gatling or if you want to get in touch with us for more information, contact us here.

If you want to learn more information about FrontLine, our enterprise version for Gatling, you should see our demo here

Load test your application now
gatling.io