Rapise – NeoLoad Integration

Contents

Revision

Date	Version	Author	Comment
2016-03-24	1.0	Denis Markovtsev	Conversion of a functional to load test described.
2016-04-14	1.1	Denis Markovtsev	Description of client performance monitoring.

Overview

This document describes the interface and workflow of Rapise - NeoLoad integration.

Convert Functional to Load Test

Prerequisites

- Rapise 4.1 or above
- NeoLoad 5.1 or above

Both tools must be installed on the same Windows host.

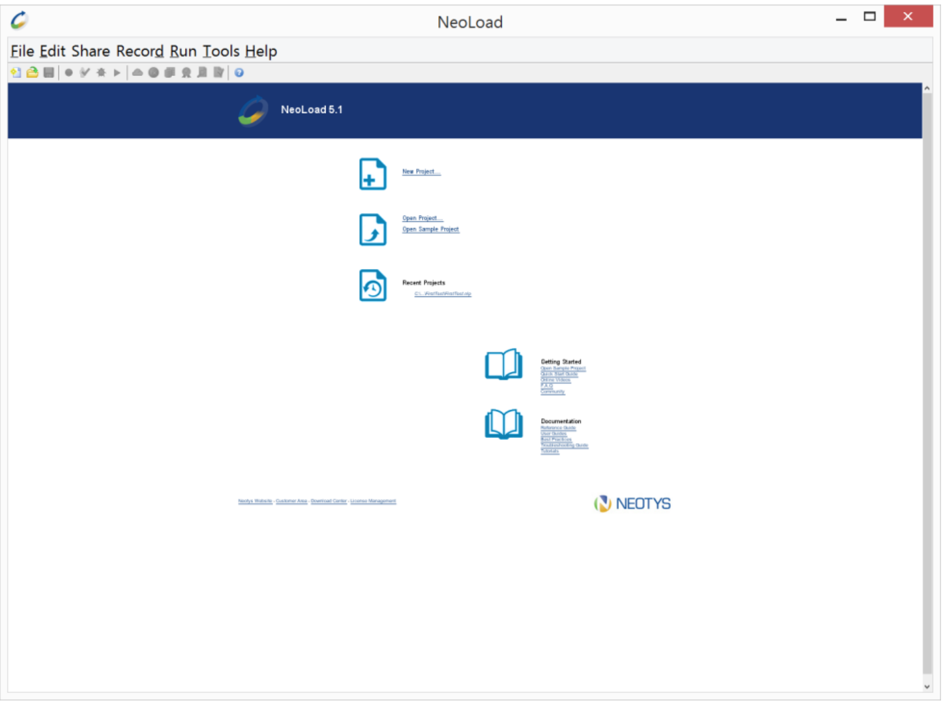
NeoLoad must have a license with Recording API enabled. The default trial version does not have this capability.

Workflow

Launch NeoLoad

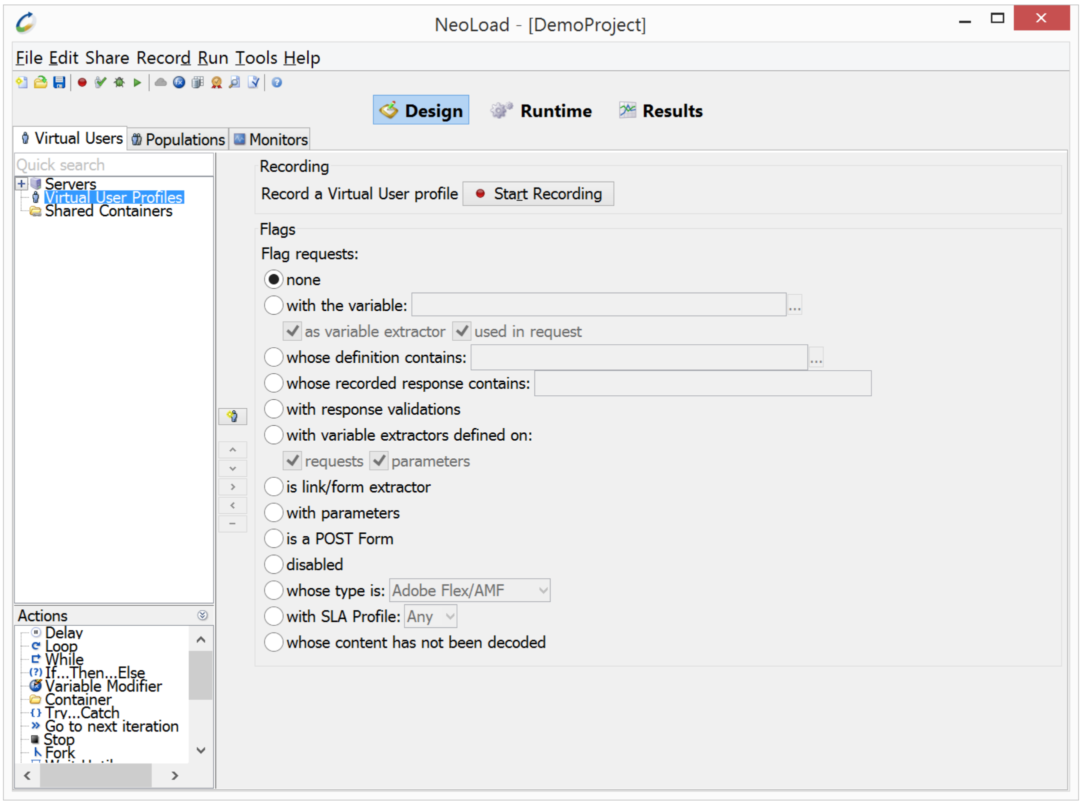
Open NeoLoad via Start menu or using a command line:

"c:\Program Files (x86)\NeoLoad 5.1\bin\NeoLoadGUI.exe"



Create or Open Existing Load Project

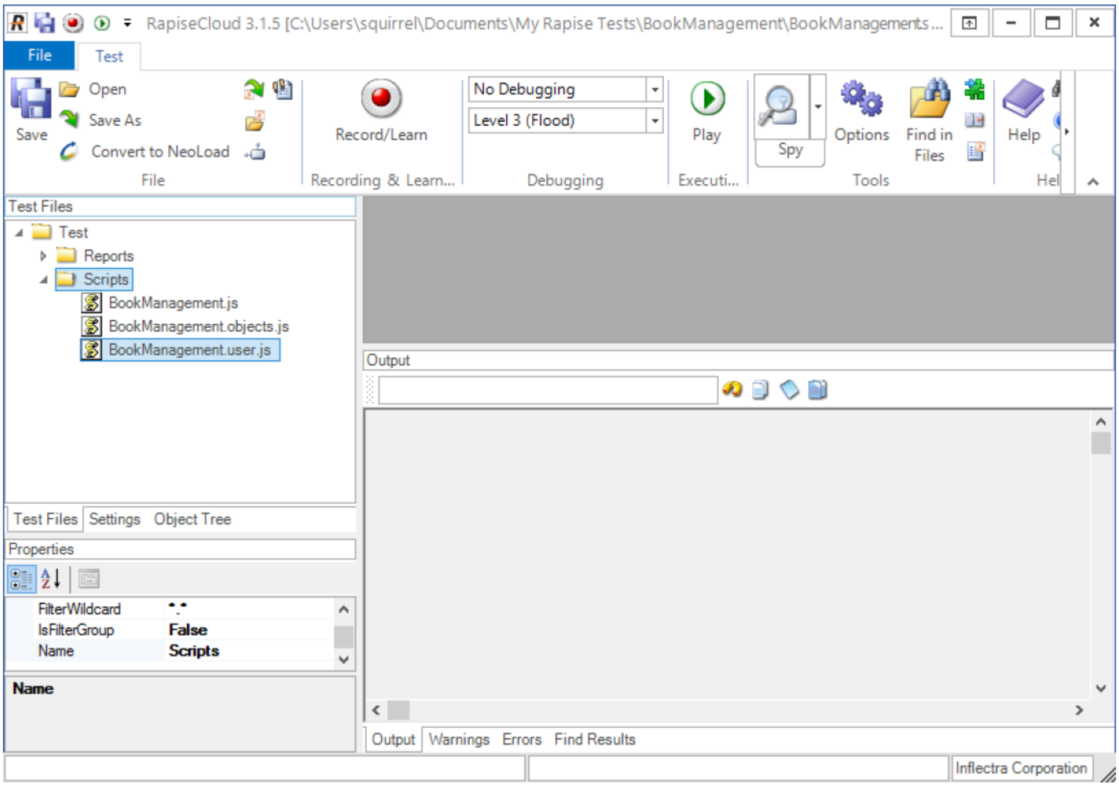
Use menu File > New or File > Open.



Flag requests:

- ● none
- ○ with the variable:
  - ☑ as variable extractor ☑ used in request
- ○ whose definition contains:
- ○ whose recorded response contains:
- ○ with response validations
- ○ with variable extractors defined on:
  - ☑ requests ☑ parameters
- ○ is link/form extractor
- ○ with parameters
- ○ is a POST Form
- ○ disabled
- ○ whose type is: Adobe Flex/AMF
- ○ with SLA Profile: Any
- ○ whose content has not been decoded

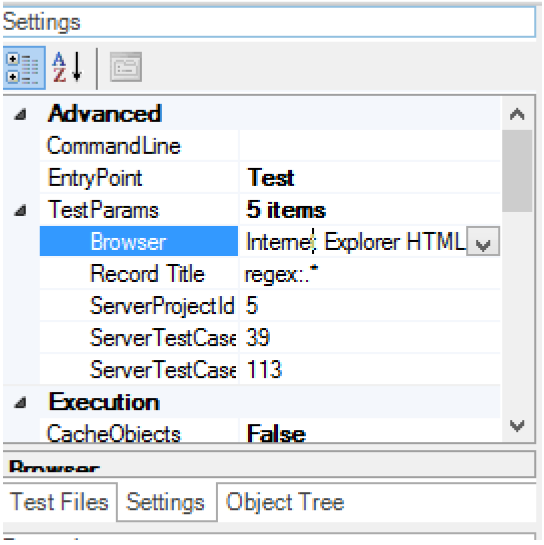
Open a Functional Cross-Browser Test in Rapise



FilterWildcard	*.*
IsFilterGroup	False
Name	Scripts

Name

Ensure that Internet Explorer HTML library is set in test parameters.



Settings

Advanced

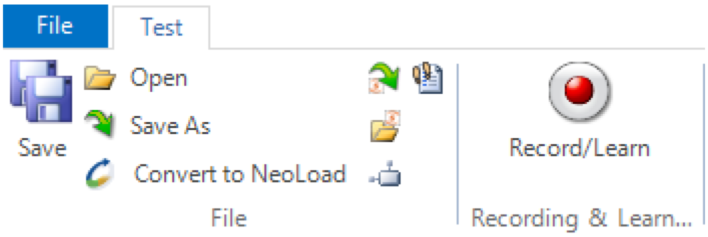
CommandLine	
EntryPoint	**Test**
TestParams	**5 items**
Browser	Internet Explorer HTML
Record Title	regex:.*
ServerProjectId	5
ServerTestCase	39
ServerTestCase	113

Execution

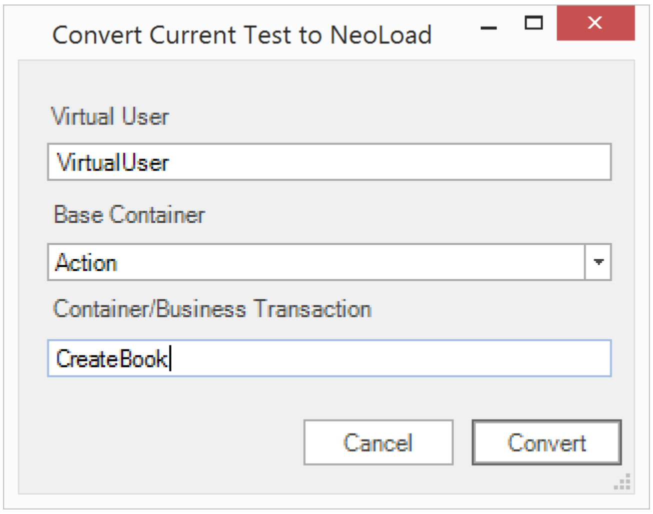
CacheObjects	**False**

Open Conversion Dialog

On the Rapise ribbon > Test tab > File group press ‘Convert to NeoLoad’ button.



On the conversion dialog set parameters:



Convert Current Test to NeoLoad

Virtual User: VirtualUser

Base Container: Action

Container/Business Transaction: CreateBook

Virtual User

Name of the virtual user to create

- Default value is "VirtualUser"
- If the name is already used, then it is automatically renamed using “_X” suffix, with X an integer incremented.
- If the name has invalid characters then they will be escaped as an underscore (_).

Base Container

Container Init/Actions/End where we want to start the record

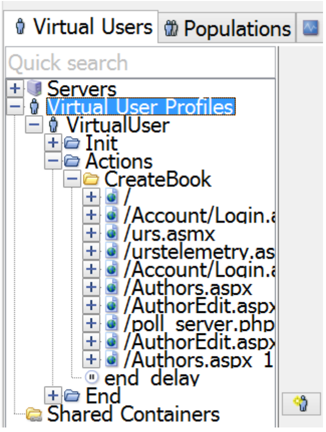
- Default is Actions.

Business Transaction

Current recording container in NeoLoad. It is just based on a single level. There is no way to specify an arborescence of containers.

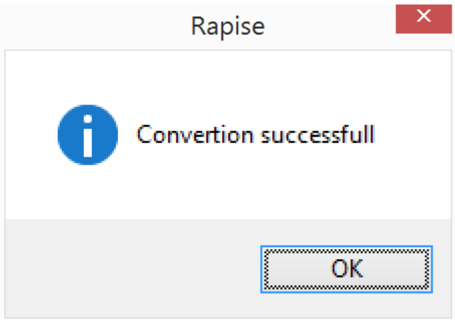
- Default is no container.
- If the name is already used then it will be made unique by adding _1, _2, etc.
- If the name is empty then no container will be used.

In the NeoLoad tree it looks this way:



Press Convert Button

After pressing Convert button Rapise will launch the test and NeoLoad will start capturing network traffic. When test playback is finished you'll see the dialog:

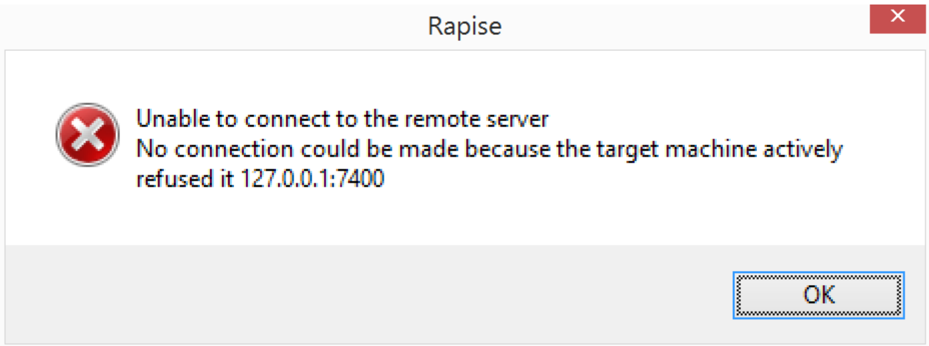


Convertion successfull

Troubleshooting

During conversion you may get a few error messages.

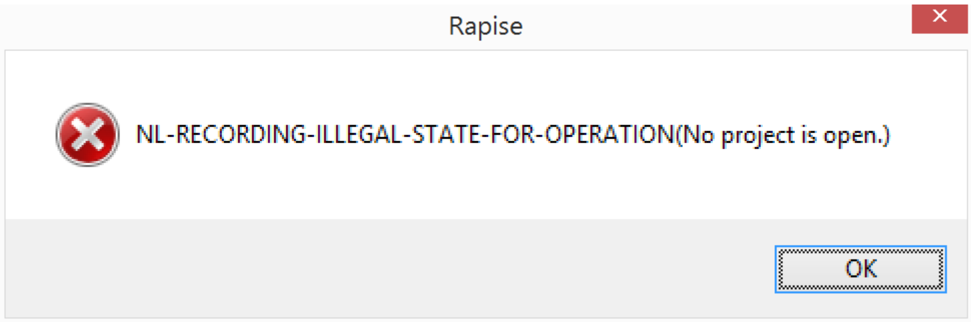
Unable to connect to the remote server



Unable to connect to the remote server
No connection could be made because the target machine actively refused it 127.0.0.1:7400

Check that NeoLoad is running.

No project is open

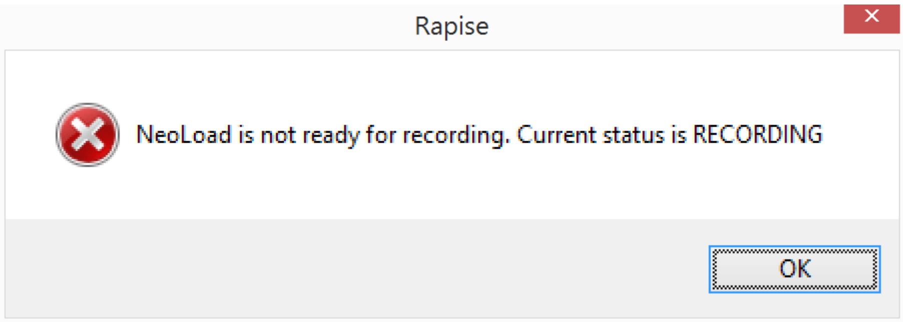


Rapise

NL-RECORDING-ILLEGAL-STATE-FOR-OPERATION(No project is open.)

Check that a project is opened in NeoLoad.

NeoLoad is not ready for recording



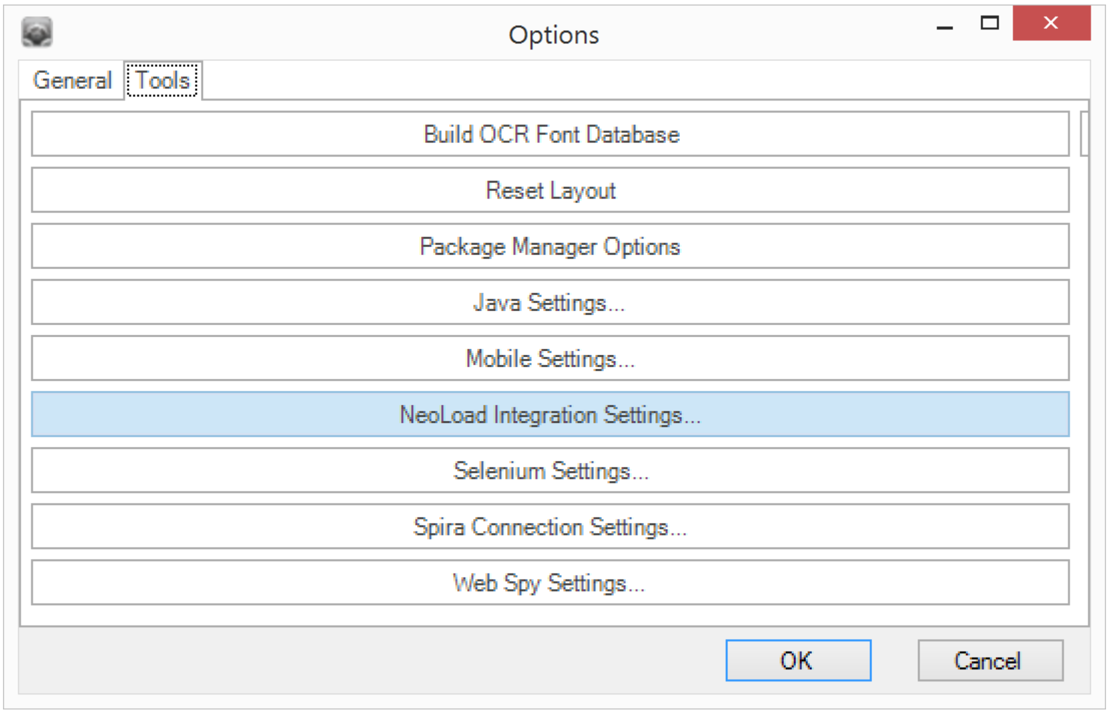
Rapise

NeoLoad is not ready for recording. Current status is RECORDING

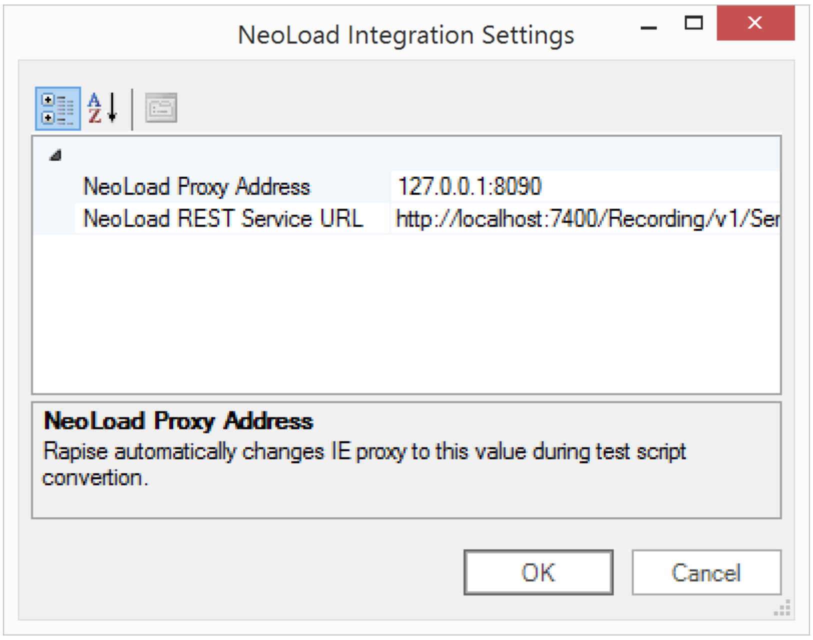
NeoLoad is in recording state. Stop recording in NeoLoad.

NeoLoad Integration Settings

There are a few global options for Rapise – NeoLoad integration. To reach them press Options in the Rapise ribbon.



Then press “NeoLoad Integration Settings…” button.



NeoLoad Proxy Address 127.0.0.1:8090

NeoLoad REST Service URL http://localhost:7400/Recording/v1/Ser

NeoLoad Proxy Address

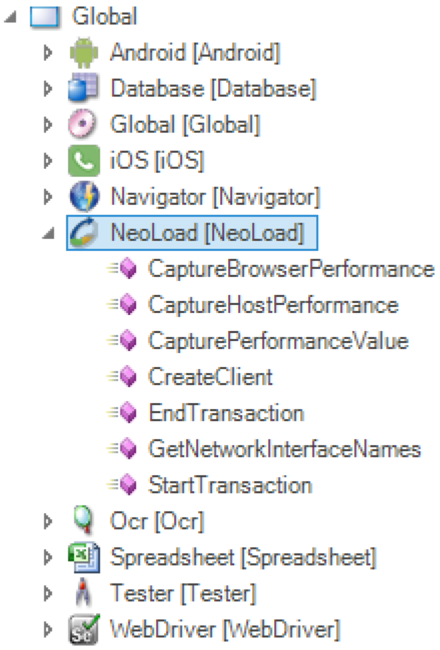
Rapise automatically changes IE proxy to this value during test script convertion.

At the current state of integration there is no any reason to change these settings.

Client Performance Monitoring

NeoLoad Global Object

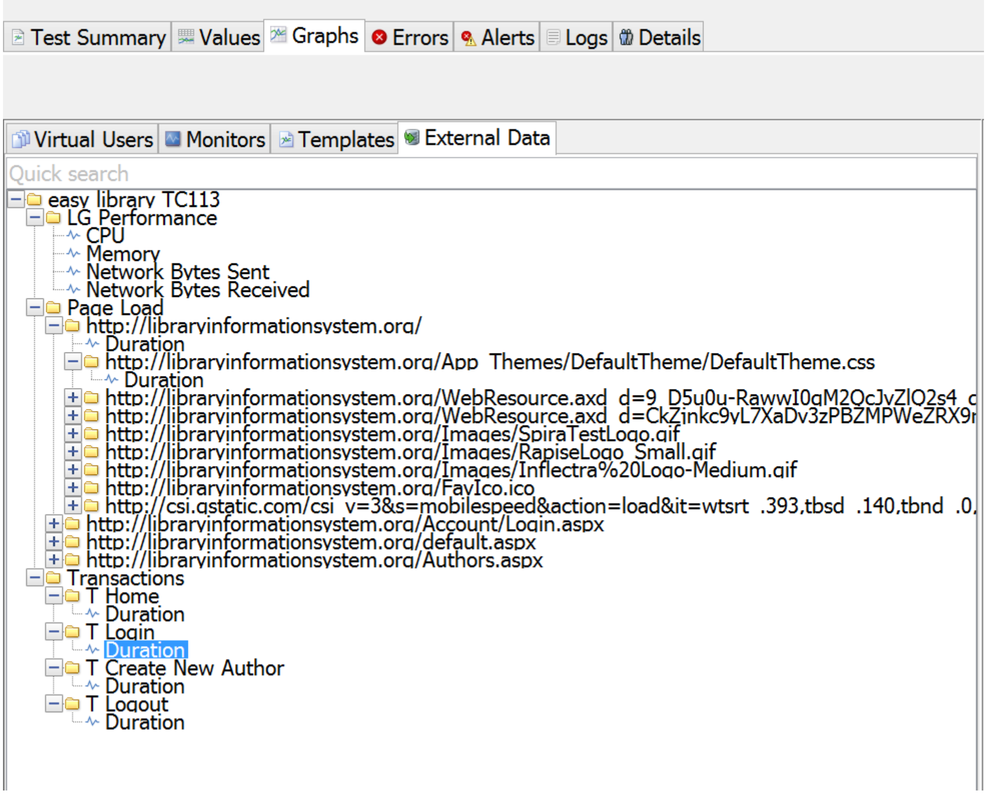
Rapise has a global object named NeoLoad which is capable of sending client performance metrics to the NeoLoad controller.



```
Global
    Android [Android]
    Database [Database]
    Global [Global]
    iOS [iOS]
    Navigator [Navigator]
    NeoLoad [NeoLoad]
        CaptureBrowserPerformance
        CaptureHostPerformance
        CapturePerformanceValue
        CreateClient
        EndTransaction
        GetNetworkInterfaceNames
        StartTransaction
    Ocr [Ocr]
    Spreadsheet [Spreadsheet]
    Tester [Tester]
    WebDriver [WebDriver]
```

NeoLoad displays received data in a tree and graph views.

Tree View



```
easy library TC113
    LG Performance
        CPU
        Memory
        Network Bytes Sent
        Network Bytes Received
    Page Load
        http://libraryinformationsystem.org/
            Duration
            http://libraryinformationsystem.org/App Themes/DefaultTheme/DefaultTheme.css
                Duration
            http://libraryinformationsystem.org/WebResource.axd d=9 D5u0u-RawwI0gM2QcJvZlQ2s4 c
            http://libraryinformationsystem.org/WebResource.axd d=CkZinkc9yL7XaDv3zPBZMPWeZRX9r
            http://libraryinformationsystem.org/Images/SpiraTestLogo.gif
            http://libraryinformationsystem.org/Images/RapiseLogo Small.gif
            http://libraryinformationsystem.org/Images/Inflectra%20Logo-Medium.gif
            http://libraryinformationsystem.org/FavIco.ico
            http://csi.gstatic.com/csi v=3&s=mobilespeed&action=load&it=wtsrt .393,tbsd .140,tbnd .0,
        http://libraryinformationsystem.org/Account/Login.aspx
        http://libraryinformationsystem.org/default.aspx
        http://libraryinformationsystem.org/Authors.aspx
    Transactions
        T Home
            Duration
        T Login
            Duration
        T Create New Author
            Duration
        T Logout
            Duration
```

The root of the tree is the name of a script that was executed by Rapise.

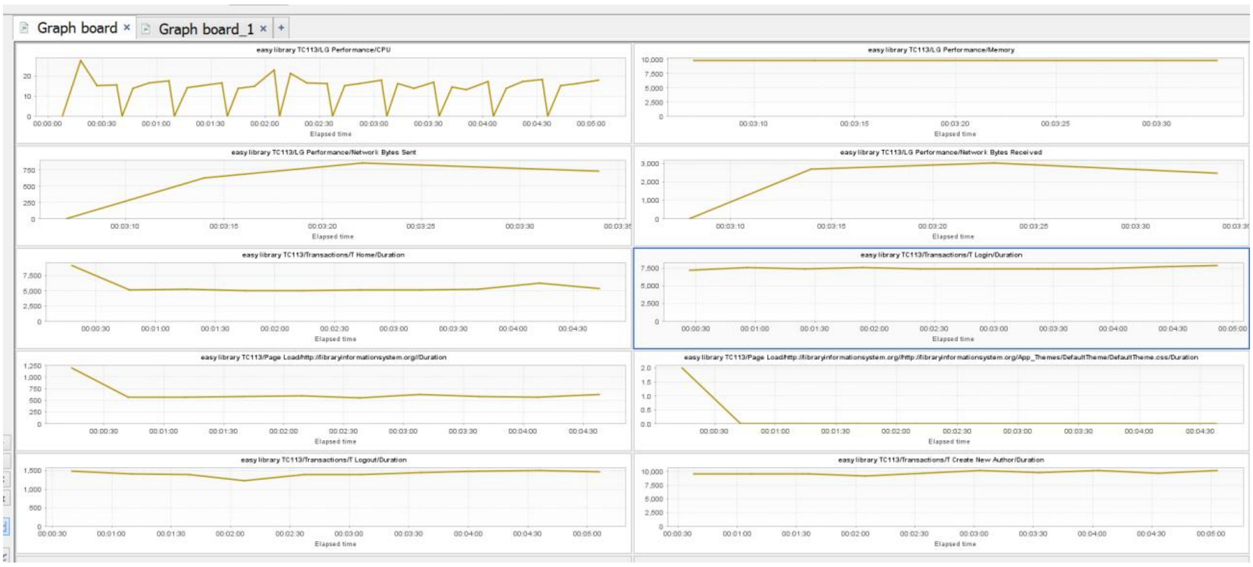
On the first level we have three groups:

- LG Performance – this is a folder of a load generator essential metrics captured by Rapise: CPU, Memory, Network Sent/Received bytes.
- Page Load – this folder contains nodes for each page and nested resource. Duration values are obtained using Window.performance structure. See also http://www.w3.org/TR/navigation-timing/#processing-model

- Transactions – this folder contains counters for transactions defined by a script.

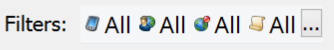
Graph View

Each leaf node can be dragged to the graph view for visualization:


Graph board ×
Graph board_1 ×
+
easy library TC113/LG Performance/CPU
easy library TC113/LG Performance/Memory
easy library TC113/LG Performance/Network Bytes Sent
easy library TC113/LG Performance/Network Bytes Received
easy library TC113/Transactions/T Home/Duration
easy library TC113/Transactions/T Login/Duration
easy library TC113/Page Load/http://libraryinformationsystem.org//Duration
easy library TC113/Page Load/http://libraryinformationsystem.org//http://libraryinformationsystem.org/App_Themes/DefaultTheme/DefaultTheme.css/Duration
easy library TC113/Transactions/T Logout/Duration
easy library TC113/Transactions/T Create New Author/Duration
Elapsed time

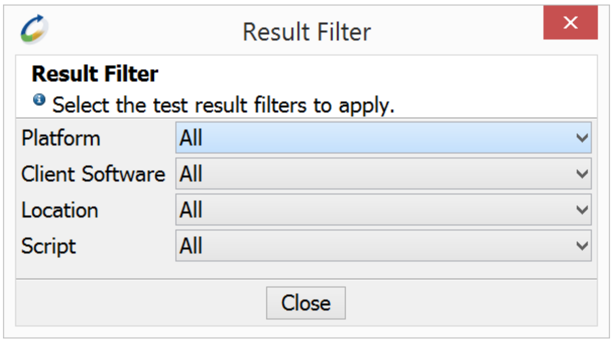
Filters

Performance results obtained from different runs can be filtered using result filter on the toolbar.



Filters: All All All All ...

Open Results Filter dialogs by pressing '…' button.



Result Filter

Select the test result filters to apply.

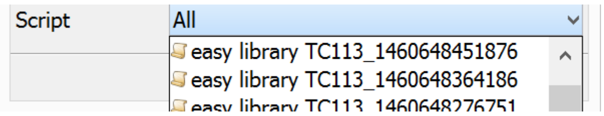
Platform	All
Client Software	All
Location	All
Script	All

Platform is a combination of hardware – OS. Hardware is passed by a user script, OS is determined by Rapise automatically. E.g. hardware1 – Microsoft Windows 8.1 Pro.

Client Software is Rapise.

Location is supplied by a script. E.g. location1.

Script is a combination of script name and time stamp. E.g.

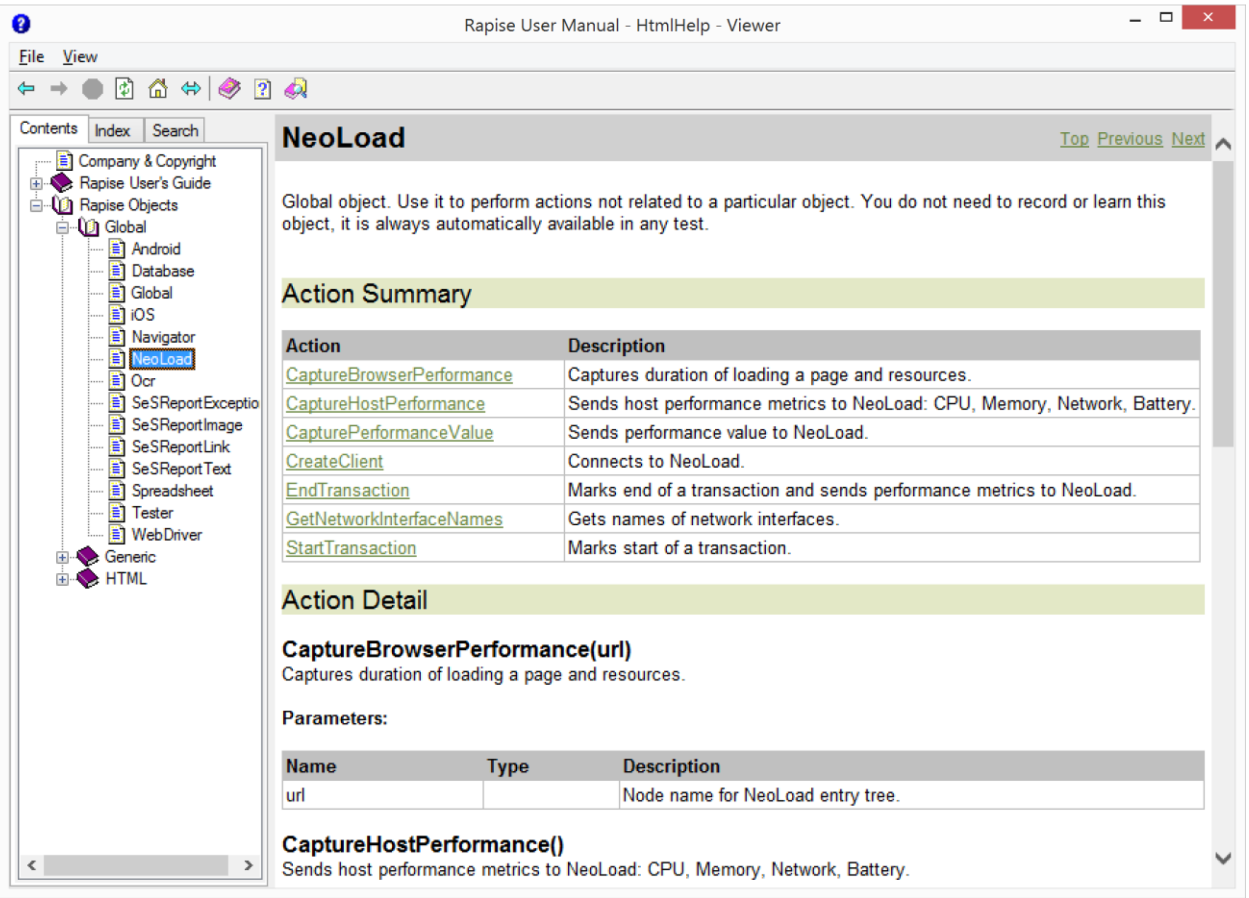


Script: All

easy library TC113_1460648451876
easy library TC113_1460648364186
easy library TC113_1460648276751

NeoLoad Object API Reference

Methods of the NeoLoad objects are described in the Rapise help file.



NeoLoad

Global object. Use it to perform actions not related to a particular object. You do not need to record or learn this object, it is always automatically available in any test.

Action Summary

Action	Description
CaptureBrowserPerformance	Captures duration of loading a page and resources.
CaptureHostPerformance	Sends host performance metrics to NeoLoad: CPU, Memory, Network, Battery.
CapturePerformanceValue	Sends performance value to NeoLoad.
CreateClient	Connects to NeoLoad.
EndTransaction	Marks end of a transaction and sends performance metrics to NeoLoad.
GetNetworkInterfaceNames	Gets names of network interfaces.
StartTransaction	Marks start of a transaction.

Action Detail

CaptureBrowserPerformance(url)
Captures duration of loading a page and resources.

Parameters:

Name	Type	Description
url		Node name for NeoLoad entry tree.

CaptureHostPerformance()
Sends host performance metrics to NeoLoad: CPU, Memory, Network, Battery.

How to Use

In order to send performance metrics to NeoLoad you need to add a sequence of calls to your script.

First, connect to the NeoLoad. When this call is executed NeoLoad must be running a script, otherwise an error will be returned.

```
NeoLoad.CreateClient("hardware1", "location1", "Broadcom 802.11ac Network
Adapter _2");
```

Let's look at the following block:

```
      NeoLoad.CaptureHostPerformance();
      NeoLoad.StartTransaction('T Home');
      NeoLoad.CaptureBrowserPerformance(SeS('Log_In').GetPageURL());
      //Click on Log In
      SeS('Log_In').DoClick();
      NeoLoad.CaptureBrowserPerformance(SeS('Username_').GetPageURL());
      NeoLoad.EndTransaction('T Home');
```

CaptureHostPerformance call sends immediate values of CPU, Memory and Network load to the NeoLoad controller.

StartTransaction/EndTransaction calls define a transaction with name ‘T Home’. Upon EndTransaction call the duration of it is sent to the NeoLoad controller.

CalculateBrowserPerformance call sends page load performance metrics to the NeoLoad controller.

There is also a general call that can send any user-defined metric to the NeoLoad controller:

```
NeoLoad.CapturePerformanceValue("First Level Folder",
"Transaction1|Duration", "milliseconds", 120);
```

Components of the second parameters can be separated by |. It allows to add more levels to the tree.