ERGONEERS

FROM SCIENCE TO INNOVATION

HOW TO...

CONFIGURE NETWORK TASK TRIGGERS?

Last Modified: 06.2018

SUMMERY

D-Lab provides the option to mark tasks or events via remote control. Defined Tasks can be triggered by an event from a simulator or by another third party app via TCP/IP connection.
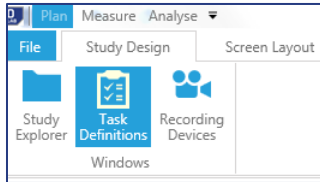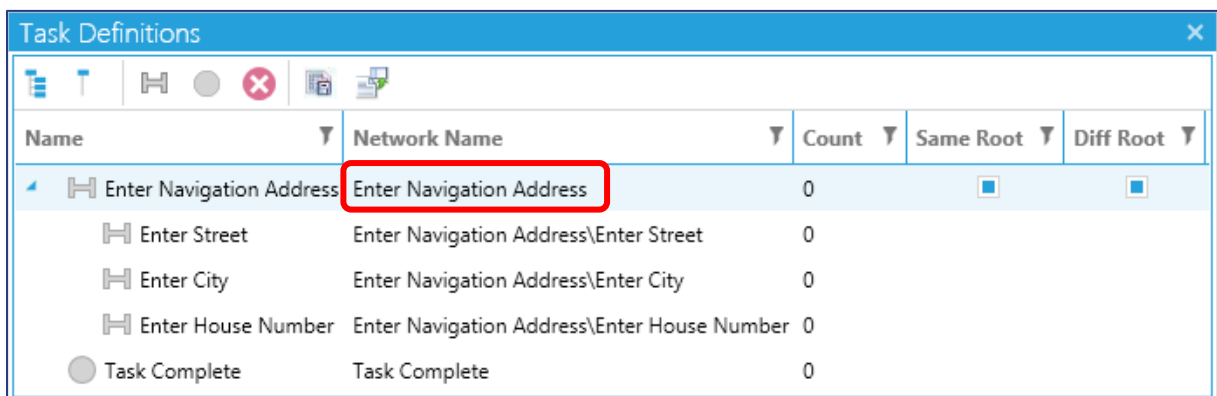
SOFTWARE REQUIREMENTS

Minimum one D-Lab 3.x Module

STEP-BY-STEP GUIDE

Define Tasks and events in D-Lab

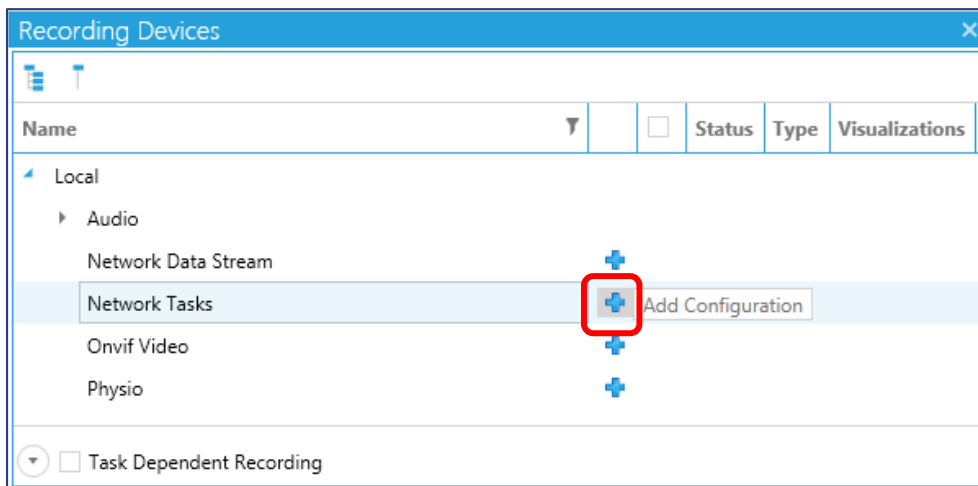1. **Define tasks** or events in D-Lab in the **Task Definitions window**.

2. A network name is automatically assigned to each task. This can be changed in D-Lab by double clicking the entry in the Network Name column. Each task must have a unique Network Name (ASCII only).
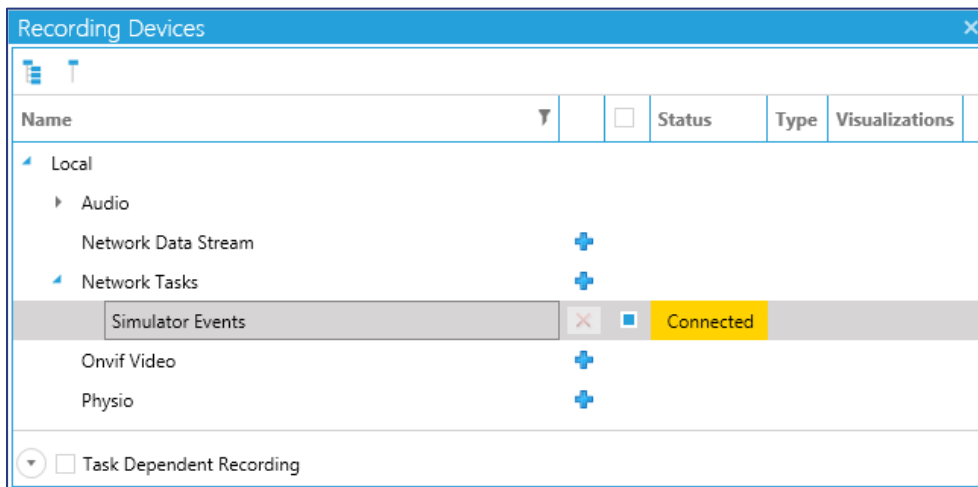

Add a Network Task Configuration

3. Create a Network Tasks Configuration in the Recording Devices window.
    <u>NOTE:</u> The D-Lab computer and the computer that sends the triggers must be in the same network.

4. Establish a TCP connection with the TCP Client (see example code below) and see the feedback in D-Lab

REQUIREMENTS TO CREATE THE TCP CLIENT

Create a TCP Client and send the network name of the task with the ending "\0" (line 50). It is possible to activate several task at once by sending names in a row (e.g. <task1 network name>\0<task2 network name>\0). For interval tasks, the first network trigger is interpreted as the start and the following one as the end. For single tasks each network trigger marks the task or event.

Code Example in C#

```
5.  using System;
6.  using System.Net.Sockets;
7.  using System.Text;
8.  using System.Threading;

9.  namespace DLab3.NetworkTrigger.Demo
10. {
11. ///
12. /// This demo shows how to send Network Task Trigger to D-Lab. If
13. /// running without parameters, it will send the trigger "TEST" to the
14. /// local instance using the default port. When using parameters, host, port,
15. /// trigger name and whether the trigger should be send in a loop can be specified.
    /// Remember to add and configure the corresponding triggers in D-Lab.
16. ///
17. public class Program
18. {
19. public static void Main(string[] args)
20. {
21. // default values
22. var host = "127.0.0.1";
23. var port = 9000;
24. var trigger = "TEST";
25. var loopDelay = 1000; // value in ms, 0 means no loop

26. // if arguments are provided, use them
27. if (args.Length == 4)
28. {
29. host = args[0];
30. port = int.Parse(args[1]);
31. trigger = args[2];
32. loopDelay = int.Parse(args[3]);
33. }
```

```csharp
34. Console.WriteLine("Demo started. Using these arguments:");
35. Console.WriteLine($" Host: {host}");
36. Console.WriteLine($" Port: {port}");
37. Console.WriteLine($" Trigger: {trigger}");
38. Console.WriteLine($" Loop delay: {loopDelay}");

39. // We need a TCP client. Important: No buffering!
40. using (var tcpClient = new TcpClient { NoDelay = true })
41. {
42. tcpClient.Connect(host, port);
43. Console.WriteLine($"Connected to {host}:{port}.");
44. using (var stream = tcpClient.GetStream())
45. {
46. // If requested we send the trigger every second until a key is pressed
47. do
48. {
49. // We just send the trigger name ASCII encoded with a trailing null terminator
    ("0")
50. var data = Encoding.ASCII.GetBytes(trigger + "\0");
51. stream.Write(data, 0, data.Length);
52. Console.WriteLine($"Sended '{trigger}'.");
53. if (loopDelay > 0) Thread.Sleep(loopDelay);
54. } while (loopDelay > 0 && !Console.KeyAvailable);
55. }
56. }
57. }
58. }
59. }
```

GET HELP

You can find answers to your questions, as well as instructions and troubleshooting information, in Ergoneers FAQ (http://www.ergoneers.com/faq).

In case of other questions or hardware problems please contact support@ergoneers.com.