

# Simple Connectivity Checker

## User Guide

**Simple Connectivity Checker** is simple and reliable package for easy checking of internet connectivity in your Unity games, with very simple implementation for both programmers and non-programmers and for everyone familiar with Unity inspector functionality. This package is cross platform solution and it works both for WiFi connectivity as well as for network carrier connectivity.

## Content

1. Features
2. Adding package to Unity project
3. Package Setup in your Unity project
4. Package Usage in your Unity project
  - 4.1. How to use the package in inspector (Simplified usage)
  - 4.2. How to use the package in code (Advanced usage)
5. Last words

### 1. Features

Simple Connectivity Checker is equipped with few quality-of-life features in order to simplify implementation and adapt to different games and development styles.

- **Unity Menu (Connectivity Checker)**
  - **Add ConnectivityManager:** This menu button adds the manager game object to your currently opened scene
  - **Toggle Connectivity in Editor:** This menu button toggles fake connected / disconnected feature to quickly check Simple Connectivity Checker implemented events in-editor and avoid real connect / disconnect of your network while

developing.

**Note:** *This feature affects only this package and is for use only while developing. Not intended to replace real connect / disconnect states.*

- **Guide:** This menu button opens this guide in your browser.

- **Singleton**

- This package is created with Singleton pattern and is persistent through all the scenes once added to scene. Recommended: Add the game object in first scene of your game and it will be available in every other scene opened.

- **ConnectivityManager Inspector**

- **Print Debug Messages (toggle):** This toggle enables / disables debug messages from ConnectivityManager script.

**Note:** *For optimization purposes disable this toggle when making production builds.*

- **Start On Awake (toggle):** This toggle enables / disables if the ConnectivityManager should start checking connectivity on Awake (recommended). If this toggle is disable you will need to Start the checking in your code (advanced users only).

- **Ping URL:** This is the URL which will be pinged by ConnectivityManager to check if there is internet connection.

**Note:** *For games with bigger user-base it is recommended to change this URL to your own webpage.*

- **Ping Interval:** This is the interval at which ConnectivityManager will ping the above URL to check for internet connection. The connectivity check is continuous and this interval should be tweaked for best optimization of responsiveness and performance. Recommended interval is between 3 – 5 seconds, but you can test

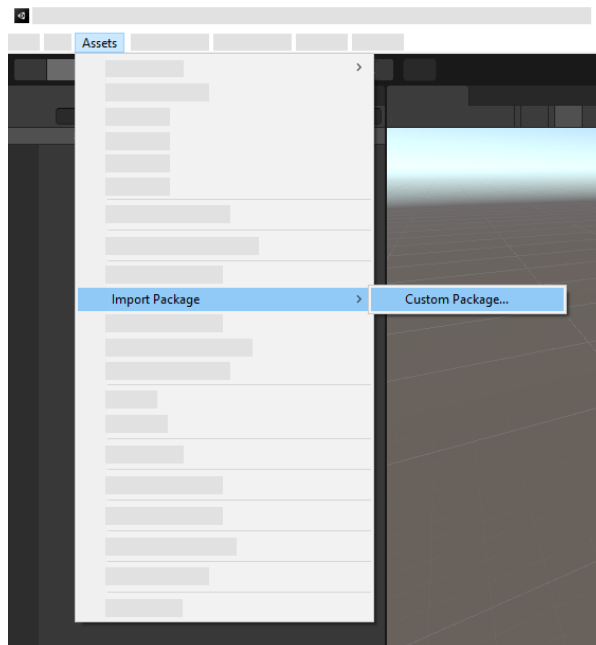
and tweak as you wish.

- **On Connectivity Change:** This event is called whenever connectivity status is changed, either from connected to disconnected or disconnected to connected. Add listeners to this even to get notified by the **Simple Connectivity Checker** when the internet status is changed. For simplified usage you can add listeners directly in inspector. For advanced usage you are also able to register and un-register to this event in code.
- **Demo Project:**
  - This package contains demo project which can be used as guide for your own implementation. The demo contains Scene with couple of buttons to show the capabilities of Simple Connectivity Checker and also code implementation of the User Interface.

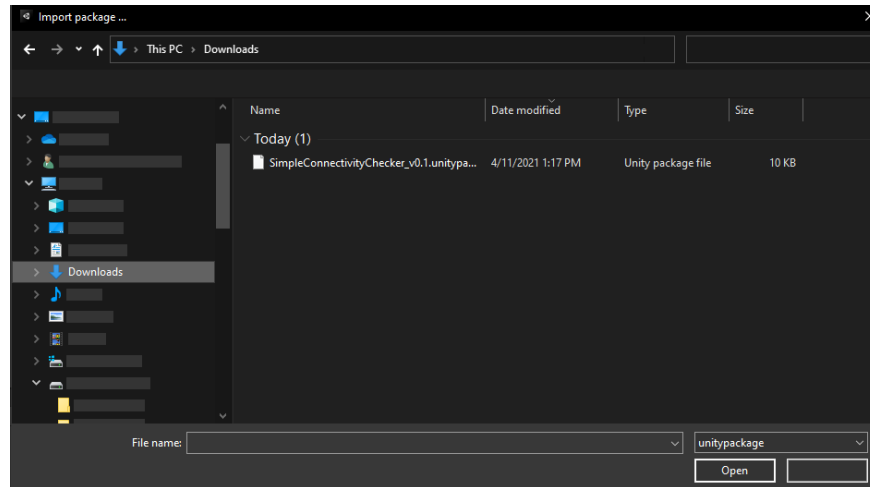
## 2. Adding package to Unity project

Those familiar with importing custom packages in Unity Projects can skip this step from the User guide.

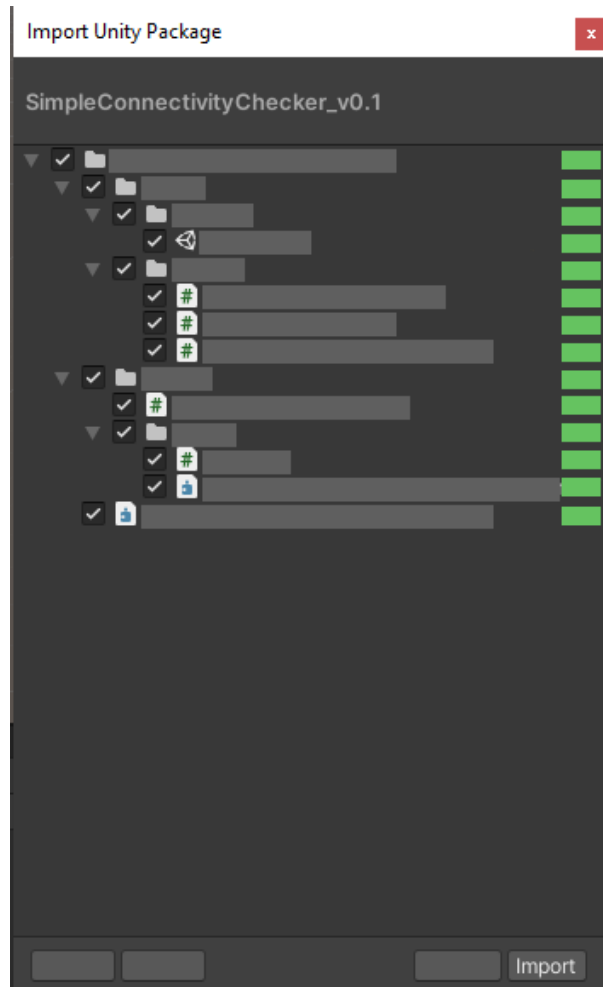
- Click on unity menu Assets > Import Package > Custom package



- Find the download location of **Simple Connectivity Checker** package
- Select **Simple Connectivity Checker** package and click Open



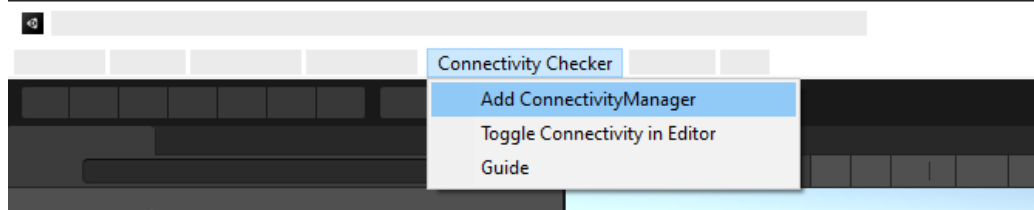
- When Import Unity Package screen shows click Import



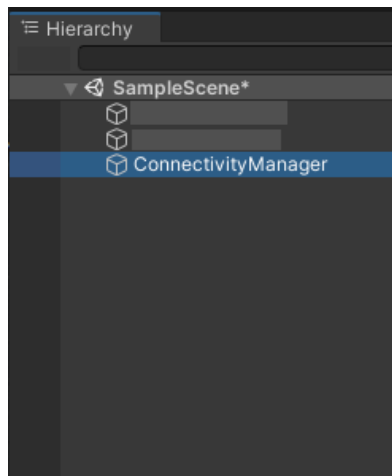
### 3. Package Setup in your Unity project

Simple Connectivity Checker features are simple and easy to setup in few steps:

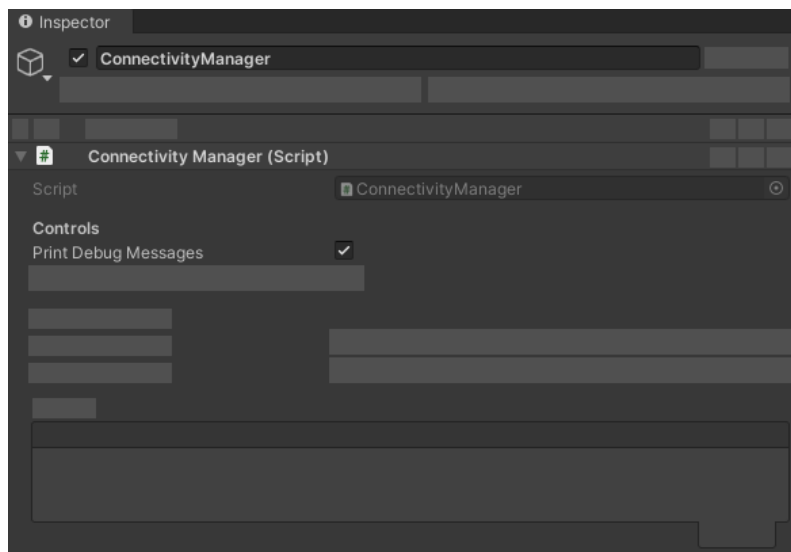
- Click on unity menu Connectivity Checker > Add ConnectivityManager



- ConnectivityManager game object will be added to your hierarchy and selected automatically



- Enable Print Debug Messages to be able to see the debug messages from Connectivity Manager in Unity Console



- Save scene changes

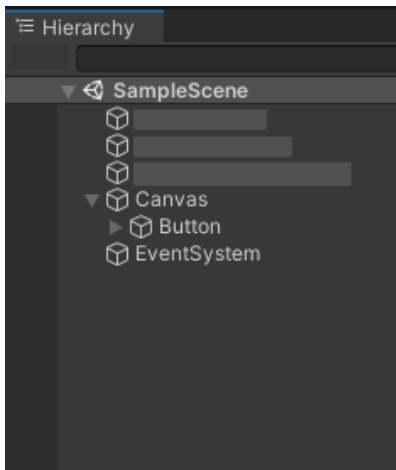
## 4. Package Usage in your Unity project

- **How to use the package in inspector (Simplified usage)**

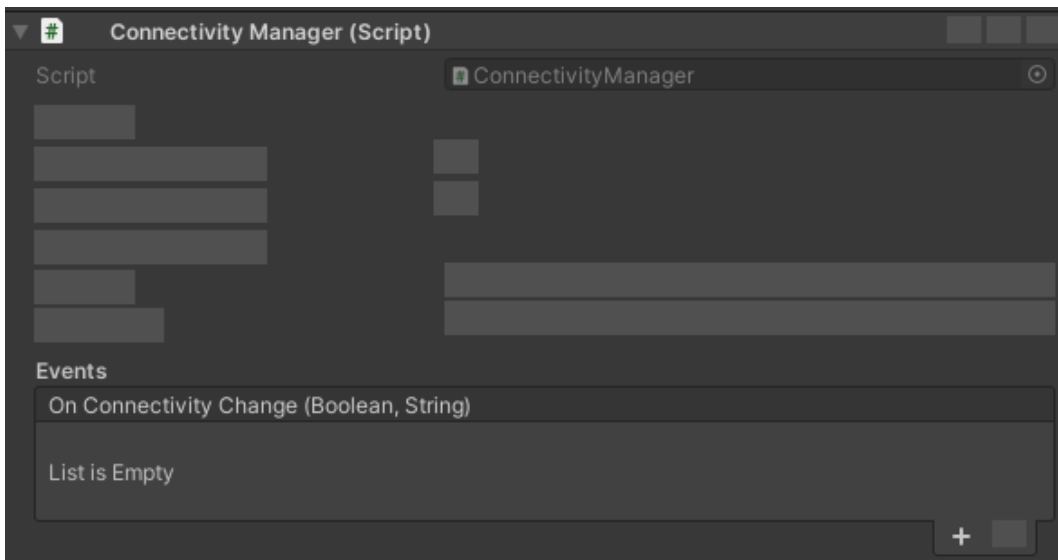
If you are not programmer and don't know how to use code, you can use this package through ConnectivityManager game objects inspector.

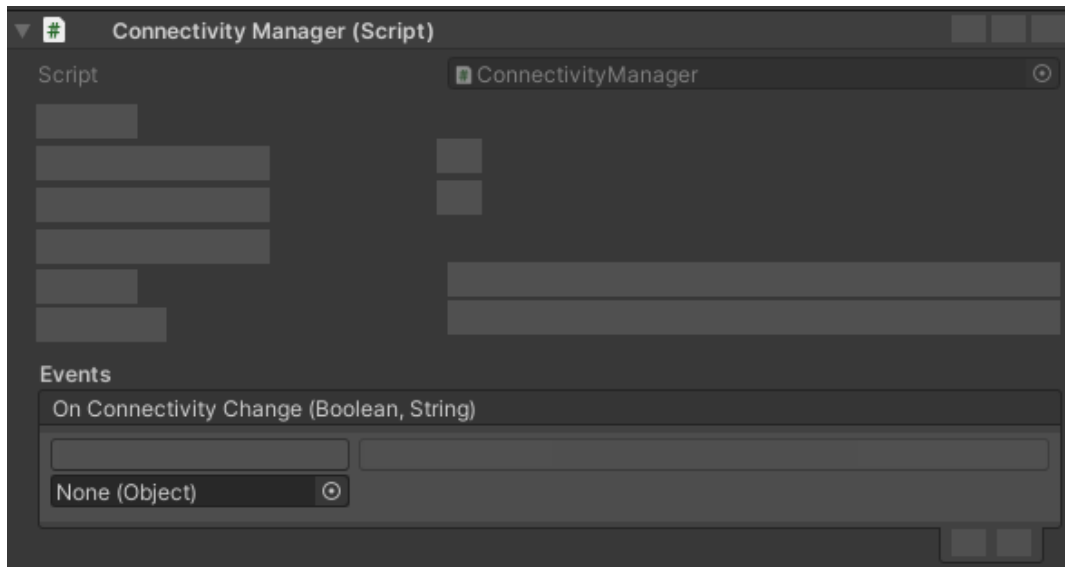
The inspector for this script exposes **On Connectivity Change** event which you can add game object or script to the event and listen for connectivity changes and execute methods for that game object. For example:

- Create a button in your hierarchy

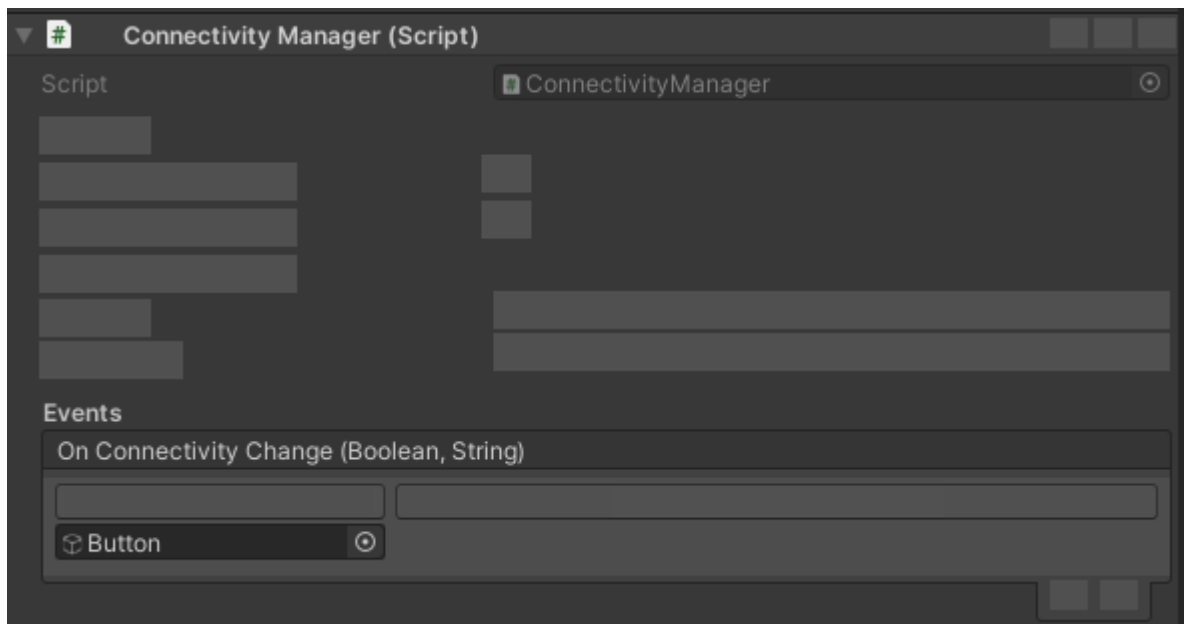


- Add Listener in **On Connectivity Change** event in **Connectivity Manager** script

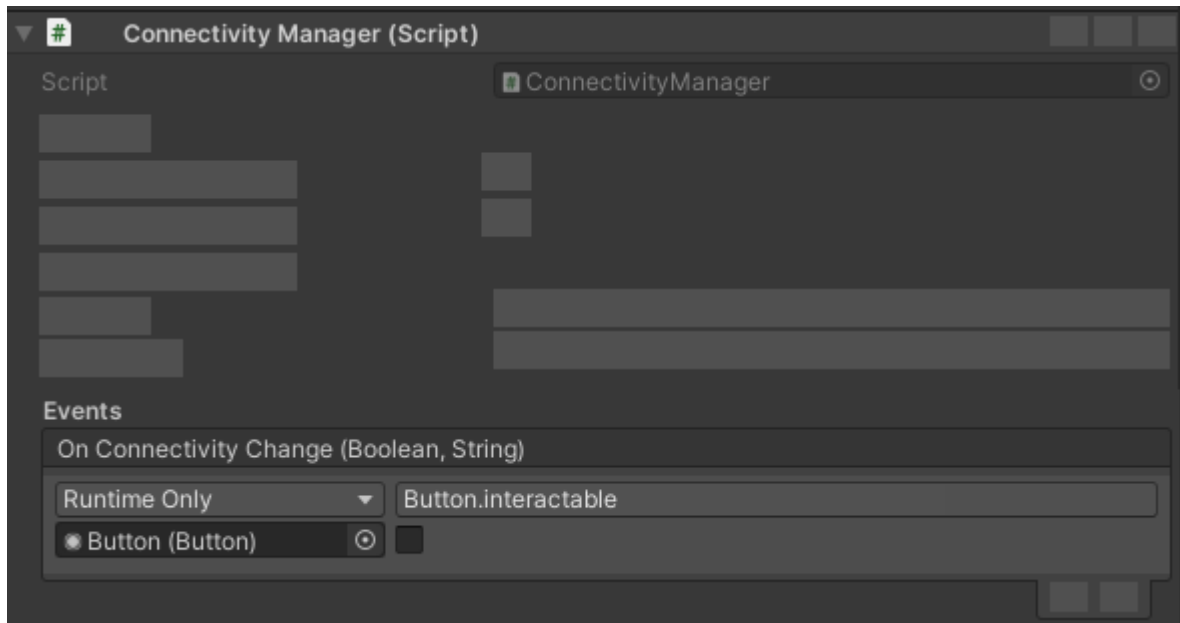




- Drag and drop the button game object to **On Connectivity Change** event on **ConnectivityManager** script in inspector.



- Select which method you want to be listener for the event. For this example, I have selected **Button.interactable** to be set to **false**, this will make the button not interactable when the status changes. If you have your own script on the button, you can add custom method to handle this event through the inspector.



- **How to use the package in code (Advanced usage)**

Advanced usage is for developers which understand programming, and gives a lot more power and control over the **Simple Connectivity Check** package.

Public interface of **Connectivity Manager** script:

- **IsTestingConnectivity**: Use this property to check if connectivity check is running or it was stopped.
- **IsConnectedDebugToggle**: This property works only in editor and is used to toggle fake connect / disconnect. By setting this property to false you can do fake disconnect and other way around.



- **IsConnected:** This property should be used to check if there is internet connectivity or not. If there is internet connection this property will return **true**, otherwise false.
- **AddConnectivityListener (UnityAction<bool, string> onConnectionStatusChange):** Use this method to register to **On Connectivity Change** listener and receive events when the connectivity status changes. For example, this event will be sent when connection changes from ON to OFF and the Unity Action you have provided will be called with bool = false (connection off) and a string message with the reason of disconnection (error message).
- **RemoveConnectivityListener (UnityAction<bool, string> onConnectionStatusChange):** Use this method to un-register from **On Connectivity Change** listener when you don't need to listen for events anymore or **OnDestroy ()**. The UnityAction provided to this method should be the same you have provided in **AddConnectivityListener**.
- **StartConnectionCheck ():** Use this method to start connectivity check. If **StartOnAwake** is enabled in inspector, this method will be called automatically on script **Awake ()**, otherwise you need to explicitly call this method.
- **StopConnectionCheck ():** Use this method to stop connectivity check when this check is no more needed. If not stopped the check is continuous and will run every ping interval (according to settings in ConnectivityManager inspector) and will not stop by itself.

For examples of advanced usage please check the Demo scene and scripts inside this package. For production builds you should remove the Demo folder.

## 5. Last words

After so many projects which needed checking for connectivity and no free and simple solution ready for use on the Asset Store, I have decided to improve on my solution, simplify it and share it with the community. I hope this helps you improve your productivity and avoids re-doing this on every single project.

If your project already uses some other packages, tools or services which use internet connection and have their own way of checking connectivity it is recommended to use those

services over pinging website (using this package), but if you need simple check to show or hide UI elements or pair it with some other simple features, I hope this package solve your problems.