Photon - Getting Started

Author: Tobias Schweers, Lead Client Engineer, Exit Games

Revision: 9
Date: 24th of Jan 2011

Table of Contents

Exit Games GmbH, Hongkongstr. 7, 20457 Hamburg, Germany
EU Headquarters: +49 40 413596-0

Exit Games Inc., One World Trade Center, 121 SW Salmon St, Suite 1100, Portland, OR. 97204

www.exitgames.com

1 Overview: Purpose of the Photon Server SDK

The Photon Server SDK enables you to develop server-side applications within the Photon Socket Server and implement logic, persistency, cheat prevention or prediction in a central instance between any number of Photon clients.

2 Documentation

2.1 Online

The most current documentation for Photon development can be found online in the **Developer Network**: http://developer.exitgames.com

To ask questions and get in touch with other Photon developers, visit our **Developer Forum**: forum.exitgames.com.

2.2 In this SDK

Aside from this file, the doc folder includes several other documents:

- **release_history_photon.txt**
  Change history for all releases of the Photon SDK. Helpful while updating.
- **photon-configuration.pdf**
  Extensive settings description which also lists optional parameters for Photon.
- **photon-perfcounter.pdf**
  Listing and explanation of the available PerfMon counters.
- **photon-license-terms.txt**
  The terms of using Photon.
- **Photon.SocketServer.chm**
  Reference of the Photon Application class-framework. All applications are built on top of this

In the doc\applications folder, you will find the application-related material:.

- **Photon.MmoDemo.chm**
  Reference documentation for the "MMO Demo". This application implements interest management, items and many other MMO-related classes and operations.
- **release_history_mmodemo.txt**
  The change history of the MMO application.
- **NetSyncObjects.pdf**
  Tutorial on syncing objects within Unity. By Caspar Strandbygaard.

Another major part of documentation is found in the supplied code. Here, the starting point would be the Lite.sln and the classes in its projects.

2.3 Tutorials and News

We are working on a series of tutorials and videos to help you with your first steps. Each will be announced in a post in our blog: blog.exitgames.com.

To find them all and stay up to date with other news, releases and fixes, subscribe to our RSS feed.

3 System Requirements

For deployment

- Windows Server 2008 (recommended) or Microsoft Windows Server 2003
    - 64bit OS recommended
- Microsoft .NET Framework 3.5 + SP1
- UDP port 5055, TCP port 4530

For development

- Windows 7 (recommended)
- Vista or Windows XP (see remarks below)
- Microsoft .NET SDK 3.5 + SP1
- Microsoft Visual Studio 2008 or 2010
- UDP port 5055, TCP port 4530

For Flash Applications

- Listener on open TCP Port 4530
- Policy Application running on TCP port 843

For Unity Webplayer Applications

- Policy Application running on TCP port 843

For Silverlight Applications

- Listener on open TCP Port 4530
- Policy Application running on TCP port 943

There are several versions of Photon included in separate folders. Chose according to your OS:

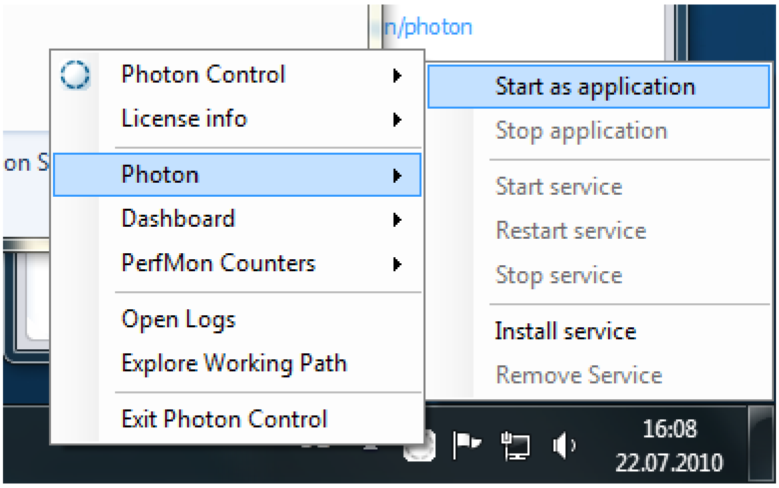
- 32bit Vista/Windows 2008 and later
    - /bin_win32
- 64bit Vista/Windows 2008 and later
    - /bin_x64 supports
- Windows XP, Windows Server 2003 and later
    - /bin_win32_xp
    - /bin_Win64_xp

4 Quickstart

4.1 Starting the Server

The server SDK includes Photon and the several precompiled applications, which it runs out of the box.

Navigate to /deploy/bin_Win32/ and *run: PhotonControl.exe*. Find the upcoming tray icon (grey Photon logo) and click on it to open the menu. Hover over “Photon” and click "Start as application".



Some OS versions will show a dialog on first start to grant access to the network. *Allow the network connection.*

Photon is now starting up. This can take a few minutes, depending on the applications that are loaded. The PhotonControl icon shows a blue circle while Photon is running.

On start, “log” folders will be created: One in the root of the deploy path and one in the “bin_*” folder. Make sure the service / process has write access to both. *Check these logs if you look out for errors*.

4.2 Testing the server

The server SDK includes a simple load testing client, which can be started from within PhotonControl.

Navigate to “Photon” and click on “Run Testclient”. This will open a console application that simulates a small number of clients that connect to your local Photon.

5 Deployment

Deployment is as easy as copying the content of the deploy folder to the target system. Photon is will run without installation.

Depending on your server hardware and OS, you need only one of the Photon binaries folders. You should also copy the bin_Tools folder, as it includes BareTail to show logs from within PhotonControl.

By convention, application binaries must be located in folders parallel to the binaries folder and must contain a "bin" folder (like: \deploy\Lite\Lite\bin\).

We centralized these files in the folder "deploy. The other folders (docs, lib, src-server) of the SDK are not required to run Photon.

Renaming any of the application-folders in deploy will invalidate the configuration, so make sure to check this when adding apps.

5.1 Licenses

Photon Licenses are provided as .license file. PhotonControl and the PhotonSocketServer.exe will read the first available .license file and use the settings from this.

To replace a license, remove old .license files and place the new one next to the PhotonSocketServer.exe that is going to run. Restart Photon and the PhotonControl.

5.1.1 Free Photon License

Photon can be run for 100 concurrent *users for free*. The license for this is not included in the SDK and must be downloaded separately.

Get it here: **http://www.exitgames.com/Download/Photon**

6 Included Applications

The Server SDK includes several applications, complete with source code. They are located in folders in sdk/src-server. Each has a different logic and here is what they do.

6.1 Counter Publisher

The Counter Publisher is a "helper application". It is used to collect and distribute PerfMon Counters of one machine to the Dashboard, which will aggregate these values into graphs.

The Counter Publisher application is started by the default configuration but is not strictly required to run.

6.2 Lite & Lite Lobby

These applications implement a basic "room based" game-server logic. Players can connect, join rooms and send messages to everyone else in a room.
The Lite Lobby extends this logic with a special room that lists existing ones. This way, players can chose which room to join.
This application is a good base for any game with smaller groups of players in separate rooms. Lite Lobby shows how to extend Lite and can be extended itself.
The Realtime Demo in client SDKs uses this application.

6.3 MMO

This server-logic implementation is geared towards bigger worlds in which a big number of players can seamlessly move and interact. The MMO application implements Worlds, Regions (parts of the world), Items and Actors (for players and NPCs). Interest areas are used to manage subscriptions of updates
The MMO folder not only includes the server source but also different demo clients.

6.4 NetSync

This application is an early implementation to synchronize objects and positional data. *It is heavily geared towards use in Unity*.

6.5 Policy

The Policy logic is a "helper application".

If Photon is not running on the same host that serves the HTML, web clients often require a policy- or crossdomain-file. The Policy Application answers requests for the policy with a suitable xml content.

Please refer to "Requirements" for a list of commonly used ports.