

C# 스크립트를 사용한 게임서버 모니터링 시스템 개발

간단하고 빠르게 게임서버 모니터링(and 원격 조작) 시스템을 만들어 보자

최흥배

<https://github.com/jacking75/choiHeungbae>



1. ScriptCS 소개 및 사용법

2. ScriptCS 활용 예로 '게임서버 모니터링 시스템 개발' 방법 소개

C++ 보다는
Java, C#, Python, Ruby, Node.js가
더 생산성이 좋다

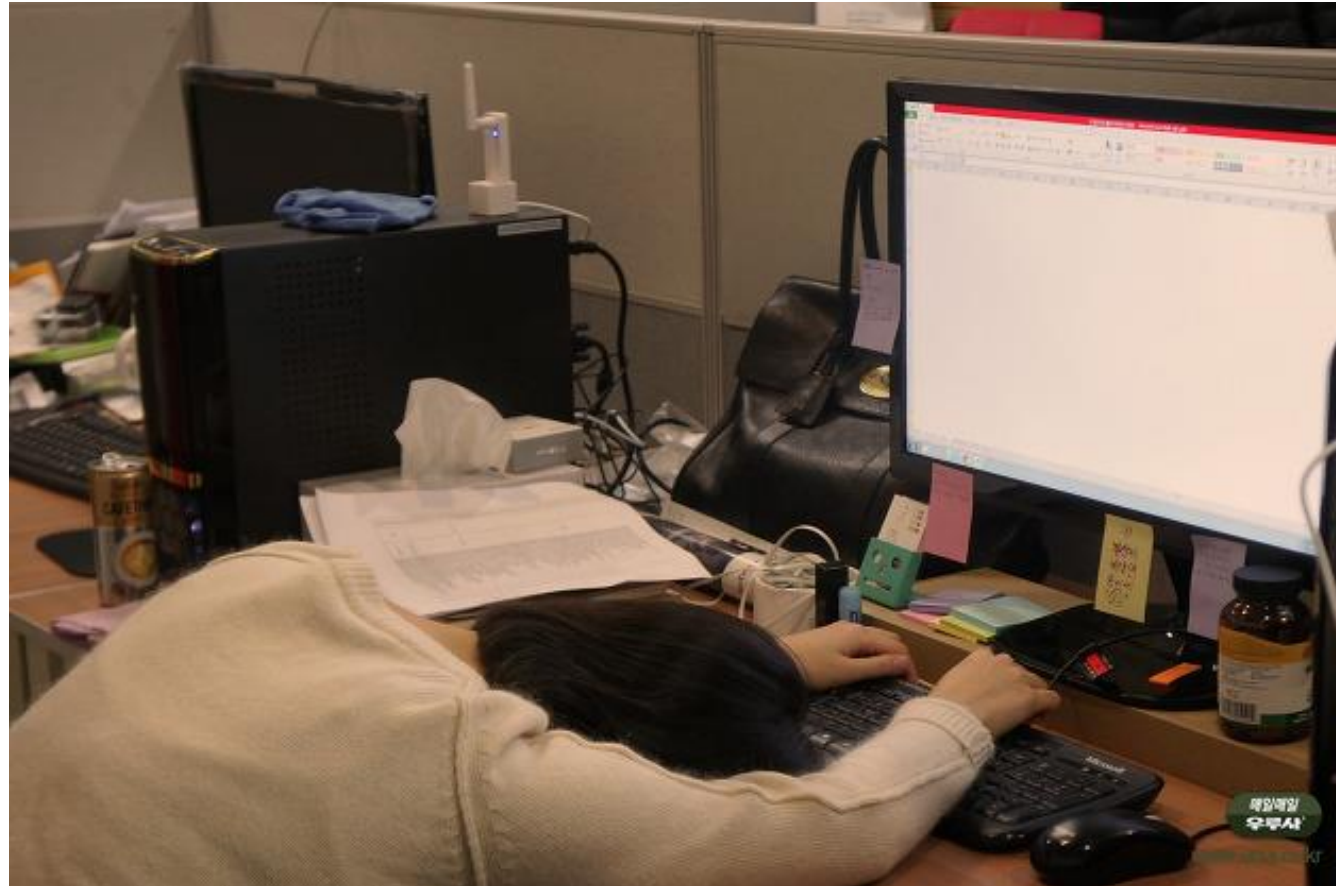
C#

- 오래 전부터 국내외에서 인하우스 툴 개발에서 사용
- C++(혹은 Java)와 비슷
- 한국 모바일 클라이언트 개발 엔진의 대세는 Unity
- Unity == C#

동적 언어 == 생산성 좋음

C# == 정적 언어

만약 C++, C#, Java만 알고 있는데..
동적 언어를 배워야 하면...
귀찮고, 시간 부족...



동적 언어 다 좋은데 문법 에러를 실행 전까지 모름

```
Python
>>> accum9M = Con("accum">9000000,1)
Runtime error <type 'exceptions.RuntimeError'>: No CellSize and Extent set
>>> env.Cellsize = 10
>>> print str(df[0].extent)
648458.46914602 5127528.79207701 735109.684115117 5190632.04328351 NaN NaN NaN NaN

>>> accum9M = Con("accum">9000000,1)
Runtime error <type 'exceptions.RuntimeError'>: No CellSize and Extent set
>>> env.Cellsize = 10
>>> accum9M = Con("accum">9000000,1)
Runtime error <type 'exceptions.RuntimeError'>: No CellSize and Extent set
>>> enc.extent = "flowdir"
Runtime error <type 'exceptions.NameError'>: name 'enc' is not defined
>>> env.extent = "flowdir"
>>> print env.extent
636299.286159372 5118437.37911638 734649.286159372 5159707.37911638 NaN NaN NaN NaN

>>> print env.cellSize
MAXOF

>>> accum9M = Con("accum">9000000,1)
Runtime error <type 'exceptions.RuntimeError'>: No CellSize and Extent set
>>>
```

**그래서 정적 언어와 동적 언어의
장점만 있으면 좋을 것 같음**

다행히 C#에 이런 것이 있음

ScriptCS



Unleash your C# from Visual Studio.

Download
ZIP File

Download
TAR Ball

View On
GitHub

What is it?

scriptcs makes it easy to write and execute C# with a simple text editor.

While Visual Studio, and other IDEs, are powerful tools, they can sometimes hinder productivity more than they promote it. You don't always need, or want, the overhead of creating a new solution or project. Sometimes you want to just type away in your favorite text editor.

scriptcs frees you from Visual Studio, without sacrificing the advantages of a strongly-typed language.

- Write C# in your favorite text editor.
- Use NuGet to manage your dependencies.
- The relaxed C# scripting syntax means you can write and execute an application with only one line of code.
- Script Packs allow you to bootstrap the environment for new scripts, further reduces the amount of code necessary to take advantage of your favorite C# frameworks.

<http://scriptcs.net/>

Getting scriptcs

Releases and nightly builds should be installed using [Chocolatey](#). Information on installing Chocolatey is available at their website.

Installing scriptcs

Once Chocolatey has been installed, you can install the latest stable version of scriptcs from your command prompt:

```
cinst scriptcs
```

Chocolatey will install scriptcs to `%APPDATA%\scriptcs\` and update your PATH accordingly.

Note: You may need to restart your command prompt after the installation completes.

Staying up-to-date

With Chocolatey, keeping scriptcs updated is just as easy:

```
cup scriptcs
```



Currently we have a very large backlog of packages in moderation and are actively taking measures to resolve it. [More...](#)

[Log On](#) [Register](#)

[Home](#)

[About](#)

[Packages](#)

[Upload](#)

[Docs*](#)

[Project*](#)

[Forum*](#)

[Shop*](#)

Let's get Chocolatey!

Chocolatey NuGet is a Machine Package Manager, somewhat like apt-get, but built with Windows in mind.

Easy Install!

To install chocolatey now, open an **administrative** cmd.exe command prompt and paste the text from the box below and press enter (or click). For

administrative PowerShell copy the second box () (Ensure Get-ExecutionPolicy is at least Bypass). If you need to install as a non-administrator or want more advanced options, see [Advanced](#).

```
c:\>choco install git
```

```
Downloading https://github.com/msysgit/msysgit/releases/download/Git-1.9.4-
Saving 10009024 of 15719536
[ooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooooo]
```

```
git.install v1.9.4
Installing git.install...
git.install has been installed.
git.install has finished successfully! The chocolatey gods have answered you
```

Chocolatey

- * Sort of like yum or apt-get but for Windows
- * Global silent installer
- * Supports multiple package feeds, including private!
- * Feeds can be local file folder shares!
- * NuGet Infrastructure + PowerShell Delivery

<https://chocolatey.org/>

Getting Started

Using the REPL

The `scripts REPL` can be started by running `scripts` without any parameters. The REPL allows you to execute C# statements directly from your command prompt.

```
C:\> scripts
scripts (ctrl-c or blank to exit)

> var message = "Hello, world!";
> Console.WriteLine(message);
Hello, world!
>

C:\>
```

Bootstrap scripts with Script Packs

Script Packs can be used to further reduce the amount of code you need to write when working with common frameworks.

- In an empty directory, install the `ScriptCs.WebApi` script pack from NuGet. The script pack will automatically imports the Web API namespaces and provides a convenient factory method for initializing the Web API host. It also replaces the default `ControllerResolver` with a custom implementation that allows Web API to discover controllers declared in scripts.

```
scriptcs -install ScriptCs.WebApi
```

```
public class TestController : ApiController {  
    public string Get() {  
        return "Hello world!";  
    }  
}  
  
var webApi = Require<WebApi>();  
var server = webApi.CreateServer("http://localhost:8888");  
server.OpenAsync().Wait();  
  
Console.WriteLine("Listening...");  
Console.ReadKey();  
server.CloseAsync().Wait();
```

- In a command prompt running as administrator, execute the `server.csx` file.

```
scriptcs server.csx
```

- Browse to <http://localhost:8888/test/> to see the result of the `TestController.Get` method.

Referencing scripts

- Move the `TestController` class from the previous example into a new file named `controller.csx` with the following content.
- On the first line of `server.csx`, reference `controller.csx` using the `#load` directive. **Note:** `#load` directives must be placed at the top of a script, otherwise they will be ignored.

```
#load "controller.csx"
```

- In a command prompt running as administrator, execute the `server.csx` file.

```
scriptcs server.csx
```

- Browse to `http://localhost:8888/test/` to see the result of the `TestController.Get` method.

Referencing assemblies

You can reference additional assemblies from the GAC or from the bin folder in your script's directory using the `#r` directive:

```
#r "nunit.core.dll"
#r "nunit.core.interfaces.dll"

var path = "UnitTests.dll";
var runner = TestSetup.GetRunner(new[] {path});
var result = runner.Run(new ConsoleListener(msg => Console.WriteLine(
    Console.ReadKey());
```

Home

[Edit](#)[New Page](#)

Adam Ralph edited this page on 5 Apr · 31 revisions

Welcome to the scriptcs wiki!

See the table of contents in the navigation side bar to the right.

► Pages 31

Table of contents

- [Home](#)
 - [Goals and Roadmap](#)
 - [Contributors](#)
 - [Who is using scriptcs](#)
 - [Installation](#)
- [Build scriptcs](#)
 - [Building on Mac and Linux](#)
- [Using scriptcs](#)
 - [The command line](#)
 - [Package installation](#)
 - [Writing a script](#)
 - [Compiling \(and caching\) to a .dll](#)
 - [Debugging a script](#)
 - [REPL](#)
 - [Running scripts in Atom](#)

<https://github.com/scriptcs/scriptcs/wiki>



Sample usages of scriptcs

🕒 50 commits

🌿 1 branch

📦 0 releases

👤 11 contributors



Branch: master ▾

scriptcs-samples / +**adamralph** Merge pull request #25 from bennage/patch-1 ...

Latest commit 96ef3e5 on 5 Apr

📁 active-directory	Added line breaks to file samples	3 years ago
📁 arcgis	Added ArcGIS sample	a year ago
📁 fluentautomation	Updated FluentAutomation sample to use Script Pack, new syntax	2 years ago
📁 itextsharp	upgraded to scriptcs 0.9 and itextsharp 5.5.0	2 years ago
📁 nancy	corrected a typo	7 months ago
📁 net	Update README.md	2 years ago
📁 nunit-runner	removing excess usings	3 years ago
📁 ravendb	removing excess usings	3 years ago
📁 servicestackhost	removing excess usings	3 years ago
📁 signalr-livereload	removing excess usings	3 years ago
📁 webapihost	reformatted webapihost code	11 months ago
📁 wpf	removing excess usings	3 years ago
📄 .gitattributes	Add .gitattributes	3 years ago
📄 .gitignore	Initial commit	3 years ago
📄 LICENSE.md	added license and updated readme	3 years ago
📄 README.md	update issue filing in readme	a year ago

<> Code

! Issues

3

🔗 Pull requests

2

📖 Wiki

📶 Pulse

📊 Graphs

Subversion checkout URL

<https://github.com>You can clone with [HTTPS](#),
[SSH](#), or [Subversion](#). ☺

📂 Clone in Desktop

📦 Download ZIP

- Install [Atom](#)

- Install [Atom Runner](#)

```
apm install atom-runner
```

- Associate csx files with scripts

```
atom
```

```
Ctrl + Shift + P
```

```
config
```

```
Enter
```

```
runner:
  extensions:
    'csx': 'scripts'
```

```
Ctrl + S
```

- Write script

```
Ctrl + N
```

```
Console.WriteLine("Hello World!");
```

```
Ctrl + S
```

```
helloworld.csx
```

- Run script

(Mac)

```
Ctrl+R
```

(Windows)

```
Alt+R
```



<https://atom.io/>

scriptcs Debugging overview

This document explains by example the steps required to use Visual Studio to debug .csx files that are executed with scriptcs. Hopefully you won't need to debug very often, but if you are in need be sure to follow this example.

Prerequisites

1. The following example shows how you can debug the [WebApiHost sample](#). This procedure assumes that you have the .csx, `scriptcs_packages.config` and `scriptcs_packages` folder already setup.
2. You must have Visual Studio 2012 or later installed.

Steps

1. Open Visual Studio.
2. Open the **Open Project** dialog by navigating to File -> Open -> Project/Solution.
3. Locate the folder where the scriptcs.exe file is located and open the executable file. Yes, you can open an .exe in VS!
4. Right-click the scriptcs solution item and click **Properties**.
5. Provide values for the following fields:
 - Arguments: `server.csx -debug`
 - Working directory: the source folder of the app you want to debug, in this the directory where server.csx is located.
6. Close the **Properties** window and save the solution.
7. Add server.csx to the solution by right-clicking the solution and selecting **Add Existing Item**.
8. Set a breakpoint in the `return "Hello World";` line of the **TestController**.
9. Press F5.
10. Open any browser and navigate to localhost:8080/api/test.

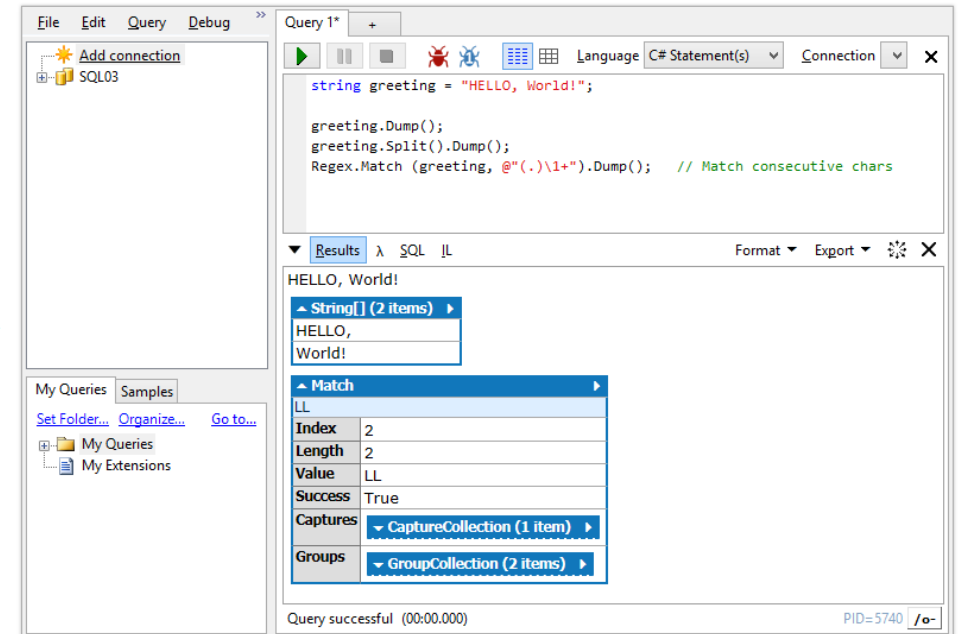
The Ultimate Scratchpad for C#, F# and VB

LINQPad is not just for LINQ queries, but any **C#/F#/VB expression, statement block or program**. Put an end to those hundreds of Visual Studio Console projects cluttering your source folder and join the revolution of LINQPad scripters and incremental developers.

Reference your own assemblies and NuGet packages. Prototype your ideas in LINQPad and then paste working code into Visual Studio. Or call your scripts *directly from the command-line*.

Experience LINQPad's rich output formatting, optional debugger and autocompletion, and the magic of dynamic development and instant feedback!

[Learn more](#)



<http://www.linqpad.net/>

Scriptcs model

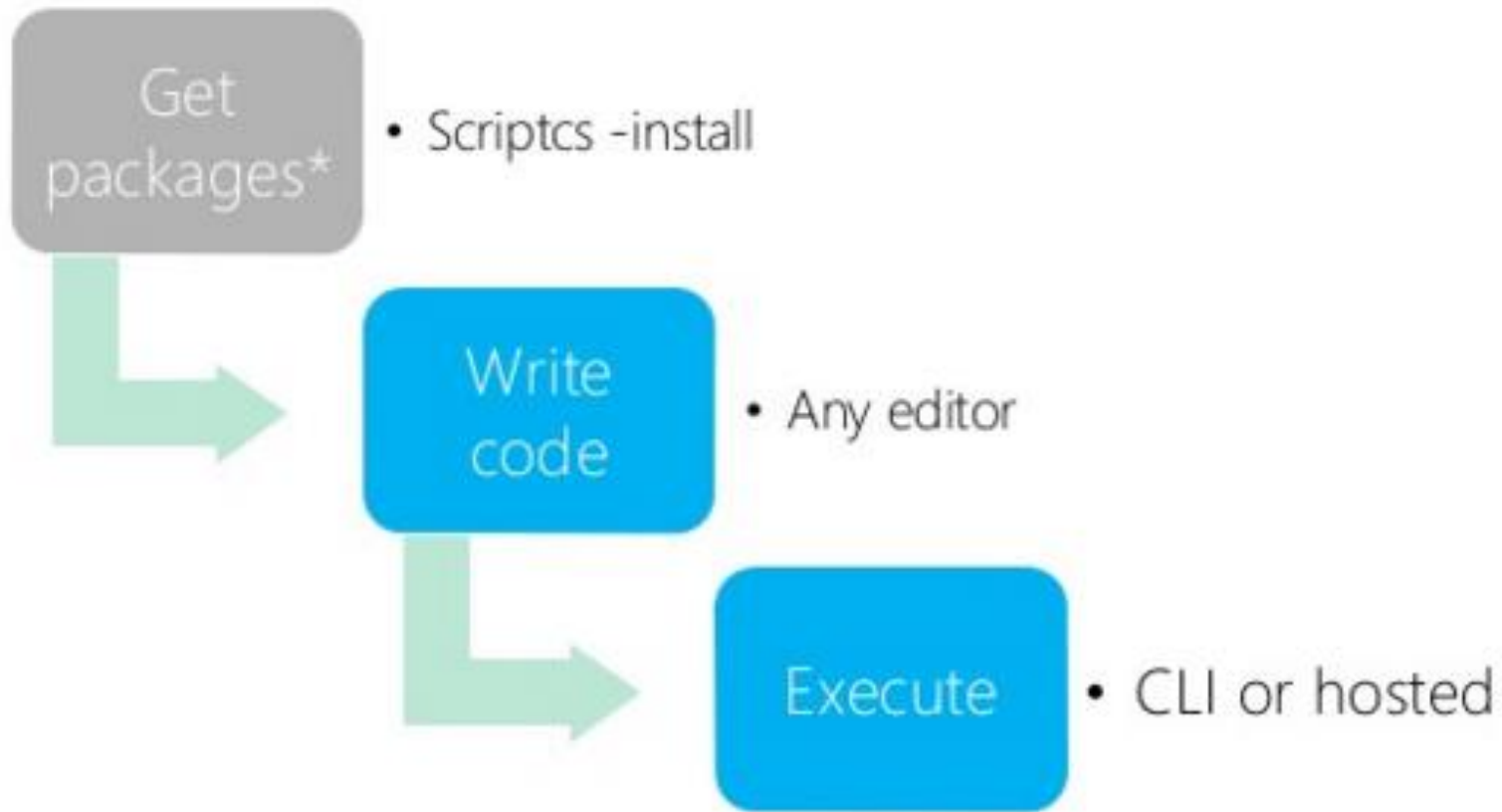
OSS libraries

MEF

Nuget

Roslyn / Mono.Csharp*

Getting started with script



<http://www.slideshare.net/FilipW/introduction-to-scriptcs/11>

ScriptPack

ScriptCs.Gui

A lightweight GUI toolkit for ScriptCS. It helps the programmer to create fully interactive forms from the REPL while keeping the REPL responsive. By [hemme](#)

- [Nuget link](#)
- [Project site](#)

```
scriptcs -install ScriptCs.Gui
```

ScriptCs.Logger.ScriptPack

Script Pack that provides a simple logger by [paulbouwer](#)

This script pack allows a log level to be set at the script level and provides the following methods to write out messages at the appropriate log level : Error, Warn, Info and Debug

- [Nuget link](#)
- [Project site](#)

```
scriptcs -install ScriptCs.Logger.ScriptPack
```

ScriptCs.Nancy

NancyFx Script Pack by [adamralph](#)

- [Nuget link](#)
- [Project site](#)

```
Install-Package ScriptCs.Nancy
```

ScriptCs.Net

A script pack that can be used to easily create TCP servers and clients. By [dschenkelman](#)

<https://github.com/scriptcs/scriptcs/wiki/Script-Packs-master-list>

ScriptPack – TCP Networking

Usage

Install the [nuget package](#) by running `scriptcs -install ScriptCs.Net`.

Creating a TCP Server

```
var net = Require<Net>();

var server = net.CreateServer(socket =>
{
    Console.WriteLine("New connection");

    socket.On(
        data: bytes => Console.Write(bytes.AsString()),
        close: () => Console.WriteLine("Connection closed"),
        error: e => Console.WriteLine("Error: {0}\r\nStackTrace: {1}", e.Message, e.StackTrace));
});

Console.WriteLine("Listening at 127.0.0.1:8080");

server.Listen(8080, "127.0.0.1").Wait();

Console.WriteLine("Closing server");

server.Close();
```

Creating a TCP Client

```
var net = Require<Net>();

var client = net.Connect(8080, "127.0.0.1", onConnect: () => Console.WriteLine("Connected to chat ro
```

<https://github.com/scriptcs-contrib/scriptcs-net>

ScriptPack – GUI

Example: creating Windows Form pop-up from the ScriptCS REPL

```
C:\git\scriptcs-gui\demo>scriptcs -install ScriptCs.Gui
INFO: Installing packages...
INFO: Installed: scriptcs.gui
INFO: Installation completed successfully.
Initiated saving packages into packages.config...
INFO: Added ScriptCs.Contracts, Version 0.8.1, .NET 4.5
INFO: Added ScriptCs.Gui, Version 0.1.0.1, .NET 4.5
INFO: Packages.config successfully created!
```

```
C:\git\scriptcs-gui\demo>scriptcs
scriptcs (ctrl-c to exit)
```

```
> using System.Windows.Forms;
> var gui = Require<Gui>();
> var f = gui.PopUp();
> f.TopMost = true;
true
> f.Sta += delegate {
var t = new TextBox();
f.Controls.Add(t);
};
{
  "$id": "1",
  "Length": 1
}
> f.Sta.Start();
> var t = (TextBox)f.Controls[0];
> t.Text = "Hello ScriptCS-GUI!";
"Hello ScriptCS-GUI!"
>
```

Building on Mac and Linux

Adam Ralph edited this page on 19 Apr · 1 revision

scriptcs now works on Mono running on Windows, Mac or Linux!

Building scriptcs from the command line

- Clone the scriptcs repo: `git clone https://github.com/scriptcs/scriptcs`
- Change to the `scriptcs` folder where the repo was cloned to.
- Checkout the dev branch: `git checkout dev`
- To make the build script executable: `chmod +x build.sh` *(if that fails prefix the command with `sudo`)*
- Run the build script: `sudo ./build.sh`.

Building scriptcs from Xamarin Studio / Mono Develop

- Clone the scriptcs repo: `git clone https://github.com/scriptcs/scriptcs`
- Checkout the dev branch: `git checkout dev`
- If you are using Xamarin Studio / Mono Develop pre 5.0, install the [Nuget Addin](#).
- Open `scriptcs.sln` in your IDE.
- Enable package restore
- Build the solution.

Running scriptcs from bash

To run scriptcs you must use the mono command, i.e. `mono scriptcs.exe start.csx`.

For convenience, add an alias in your `~/.bash_profile` for easily running scriptcs i.e:

```
alias scriptcs='mono ~/scriptcs/scriptcs.exe'
```

Roadmap - Priority in []

- [1] Validate against Packages.config - Only load packages in the config, error if packages are missing. (Filip) **DONE**
- [1] Includes - #Load don't reinvent / be consistent (Filip) **DONE**
- [1] Scriptcs API - Allows hosting Scriptcs (Justin) **DONE**
- [1] Validate binaries in scripts are present in bin.
- [1] Team city build. **DONE**
- [2] C# interactive window support (Filip)
- [2] Script packs - 3rd party plugins (Glenn) **DONE**
- [2] Export to exe
- [2] Export to VS Project
- [2] Sublime Text Plugin **DONE**



대규모 서비스 운용을 위한 선배들의 조언



Simple

모니터링

게임 서버 프로그래머가 주로 하는 일



게임 서버 개발

운영 툴, 서버 모니터링(원격 조작) 툴

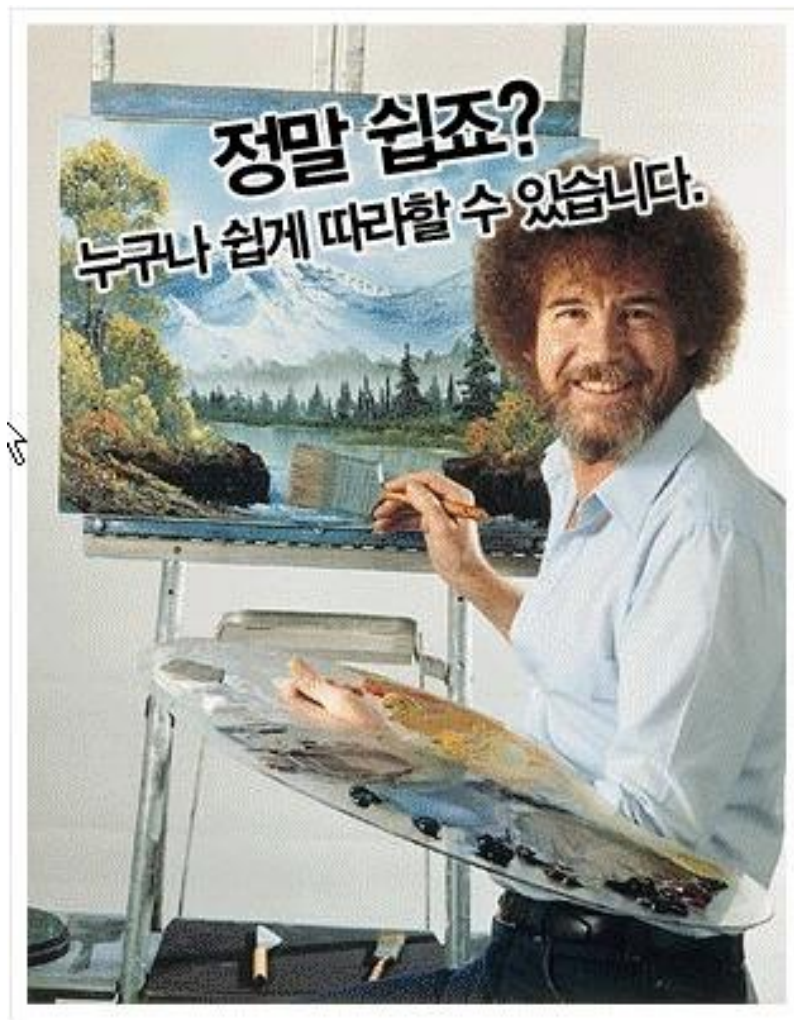
게임 개발은 바쁘다 그래서 모니터링
툴은 언제나 제일 뒤에....

그러나 개발 후반부에는 더 바쁘다

게임 서비스 전 시간이 부족하지만

빨리 만들고 and 유지보수도 좋아야

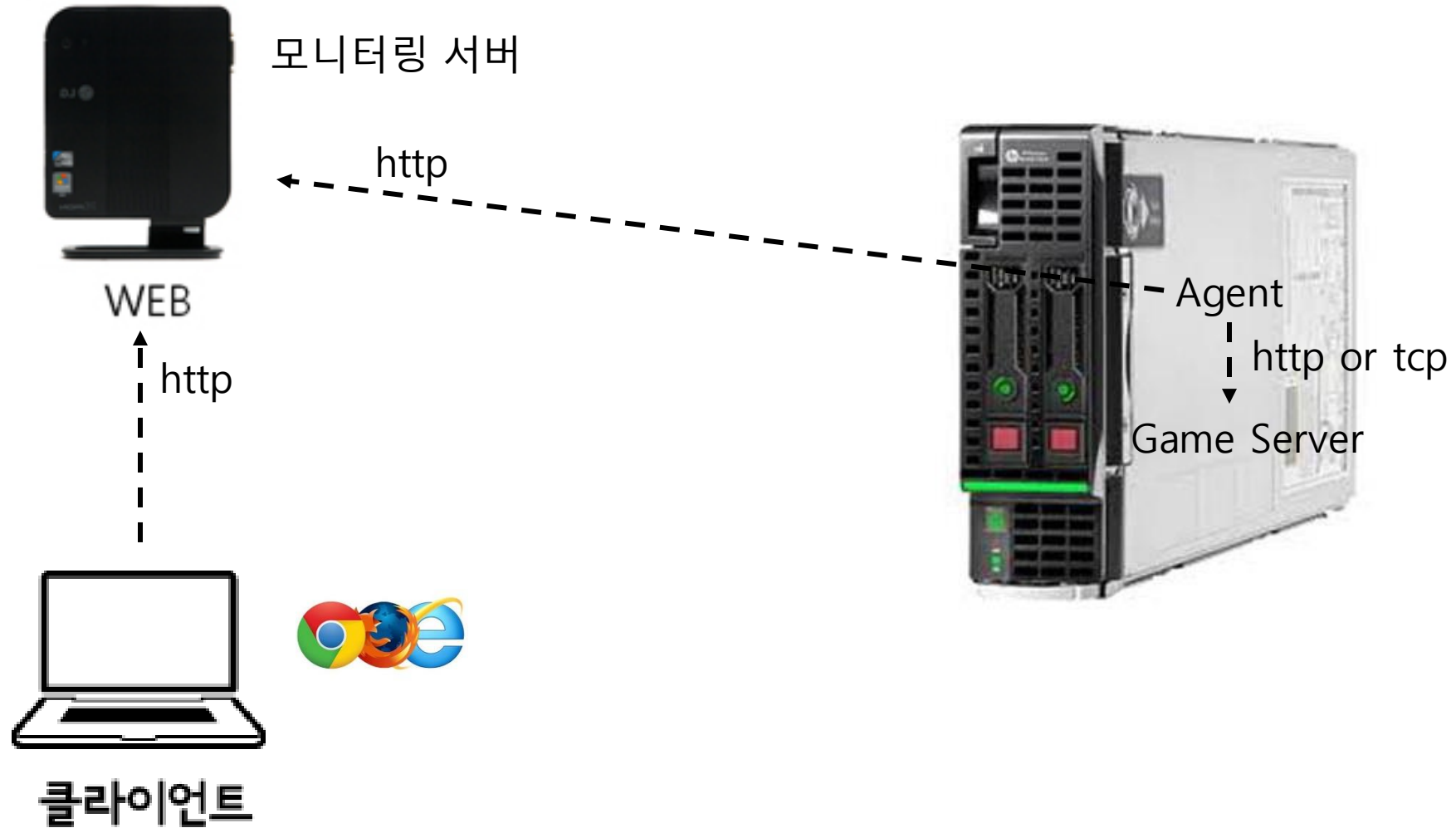
그럼 이제 만들어보죠... 게임 서버 모니터링 시스템

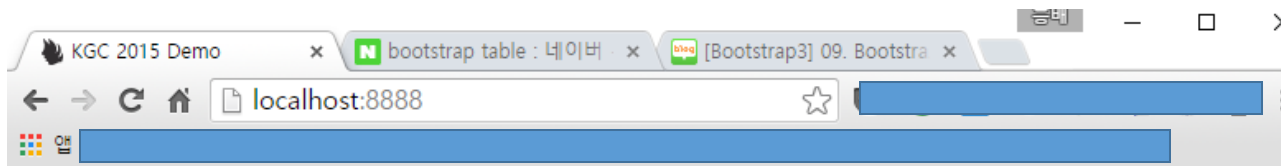


필요한 기능

- 서버 상태 표시
- 관리자 로그인
- 서버 on/off
- 서버 실행 파일 update
- (미 구현)이상 발생 시 통보하기. sms, 텔레그램, Pushbullet 연동

구조도





KGC 2015 Demo 서버 관리

오후 2:50:51 현재 로그인 중인 관리자: 없음

관리자 ID: 관리자 암호: 로그인

Agent IP	애플리케이션 이름	상태	CPU 점유율(%)	메모리 사용량(MB)
12.2.8.3	HttpGameServer	Stop	0	0
12.2.8.3	TCPGameServer	Stop	0	0

- 12.2.8.3##HttpGameServer 서버 실행 하기
- 12.2.8.3##HttpGameServer 서버 종료 하기
- 12.2.8.3##HttpGameServer 서버 패치 하기

```
관리자: 명령 프롬프트 - scriptcs server.csx -- -loglevel debug
[{"AgentIP": "12.2.8.3", "AppTitleName": "TC", "AppStatus": "Stop", "Process_CPU_Percent": "0", "All_CPU_Percent": "0", "App_Memory_MB": "0", "App_Memory_Percent": null, "Machine_Memory_Percent": "0", "Machine_Memory_GB": "0"}]
[2015-11-27T14:50:47] DEBUG called AgentDataStroage.Loop
[2015-11-27T14:50:49] DEBUG called AgentDataStroage.Loop
[2015-11-27T14:50:49] INFO AddUpdate - Update. 12.2.8.3-[{"AgentIP": "12.2.8.3", "AppTitleName": "HttpGameServer", "AppStatus": "Stop", "Process_CPU_Percent": "0", "All_CPU_Percent": "0", "App_Memory_MB": "0", "App_Memory_Percent": null, "Machine_Memory_Percent": "0", "Machine_Memory_GB": "0"}]
[2015-11-27T14:50:50] DEBUG called AgentDataStroage.Loop
[2015-11-27T14:50:52] DEBUG called AgentDataStroage.Loop
```

```
관리자: 명령 프롬프트 - scriptcs agent.csx -- -loglevel debug
관리자_명령을_처리한다. 2015-11-27 오후 2:50:44
[2015-11-27T14:50:44] DEBUG 관리서버에서_명령을_가져온다. {"Command": "", "AppServer": ""}
관리자_명령을_처리한다. 2015-11-27 오후 2:50:46
[2015-11-27T14:50:46] DEBUG 관리서버에서_명령을_가져온다. {"Command": "", "AppServer": ""}
관리자_명령을_처리한다. 2015-11-27 오후 2:50:48
[2015-11-27T14:50:48] DEBUG 관리서버에서_명령을_가져온다. {"Command": "", "AppServer": ""}
관리자_명령을_처리한다. 2015-11-27 오후 2:50:50
[2015-11-27T14:50:50] DEBUG 관리서버에서_명령을_가져온다. {"Command": "", "AppServer": ""}
관리자_명령을_처리한다. 2015-11-27 오후 2:50:52
[2015-11-27T14:50:52] DEBUG 관리서버에서_명령을_가져온다. {"Command": "", "AppServer": ""}
```

MonitoringSystem

Agent

scripts_packages

agent.csx

aws_s3_service.csx

computerStatus.csx

configuration.csx

cron_processCommand.c

cron_sendAppStatus.csx

scriptcs_packages.config

unitTest.csx

Common

defineData.csx

netHttpRequester.csx

util.csx

Server

Content

bootstrap-3.3.5-dist

jquery-2.1.4.min.js

scripts_packages

Nancy.0.23.0

Nancy.Hosting.Self.0.2

Newtonsoft.Json.6.0.8

ScriptCs.Contracts.0.9.

ScriptCs.Logger.Script

ScriptCs.Nancy.0.16.0

ScriptCs.ObjectDumper

ScriptCs.Request.1.0.0

ScriptLibraries.csx

Views

index.html

adminUser.csx

agentDataStroage.csx

scriptcs_packages.config

server.csx

server.csx

index.html

agent.csx

```
1 #load "..\Common\defineData.csx"
2 #load "configuration.csx"
3 #load "computerStatus.csx"
4 #load "cron_sendAppStatus.csx"
5 #load "cron_processCommand.csx"
6 #load "aws_s3_service.csx"
7
8 // 실행 명령어.
9 //scriptcs agent.csx -- -loglevel debug // 로그 레벨을 디버그 레벨로 한다
10 //scriptcs agent.csx -- -loglevel info // 로그 레벨을 정보 레벨로 한다
11 //scriptcs agent.csx -- -loglevel error // 로그 레벨을 에러 레벨로 한다
12
13
14 var logger = Require<Logger>();
15
16 var dumper = Require<ObjectDumper>()
17     .Compact()
18     .Build();
19
20
21 public static class Global
22 {
23     public static Logger logger;
24     public static ObjectDumper dumper;
25 }
26 Global.logger = logger;
27 Global.dumper = dumper;
28
29
30 logger.Info("스크립트 시작 ~");
31
32
33 bool IsWaiting = true;
34 main();
35
36 void main()
37 {
```

좀 더 응용해보면....

- 로그 보기
- 게임 데이터 보기/편집
- 그 외 다양한 툴



https://github.com/jacking75/kgc2015Demo_ServerMornitoringSystem