

# NETDRONE ENGINE GUIDE BOOK



## 목차

어떤 게임을 만들수 있나요?.....	2
주요 특징은 무엇인가요?.....	2
넷드론 엔진 제품 소개.....	3
시스템 요구 사항.....	3
제품 구성.....	4
상호작용 게임 구조.....	5
멀티플레이어 게임의 흐름.....	5
분산처리 서버 구성.....	6
개발 방법 알아보기.....	7
Unity Editor로 C# 클라이언트 만들기.....	7
MonoDevelop으로 C# 독립형 서버 만들기.....	8
Visual Studio로 C# 독립형 서버 만들기.....	9
Eclipse CDT로 C++ 독립형 서버 만들기.....	10
Eclipse PDT로 PHP 웹 서버 만들기.....	11
예제 및 도구 설명.....	12
서버 예제.....	12
클라이언트 예제.....	15
CSV to Binary CSV.....	18
구입 안내.....	19
인디 & 스튜디오.....	19
기업.....	19

ORIGIN STUDIO INC.

## 어떤 게임을 만들수 있나요?

네트워크 엔진이란 렌더링 및 피직스 엔진과는 성격이 다른 다수의 클라이언트 들이 실시간 상호작용을 할 수 있도록 도와주는 엔진입니다. 이를 위해 논리적인 클라이언트 및 서버를 만들게 되며 제작에 필요한 모든 것을 넷드론 엔진이 제공합니다.

넷드론은 MMO(Massively Multiplayer Online) 또는 MO(Multiplayer Online) 게임 제작에 특화된 엔진입니다. 콤팩트한 프로토콜 설계가 되어있어 MMORPG, FPS, TPS 같이 트래픽이 심한 게임에서 사용해도 쾌적한 플레이 환경을 제공할 수 있습니다.

또한 NDP(NetDrone Protocol)는 빠르고 강력한 패킷 암호화 기술을 사용하고 있어 해킹에 매우 안전합니다.

이제 Mobile 및 Console 환경에서도 PC 환경과 동일한 게임을 만들어 보세요.

## 주요 특징은 무엇인가요?

넷드론 엔진은 C# 및 C++ API 구조가 90% 이상 동일합니다. 이는 어떠한 언어로 개발하여도 동일한 구조의 설계가 가능하다는 뜻이되며 C# 프로그래머, C++ 프로그래머 상관 없이 하나의 프로젝트를 진행할 수 있음을 의미합니다.

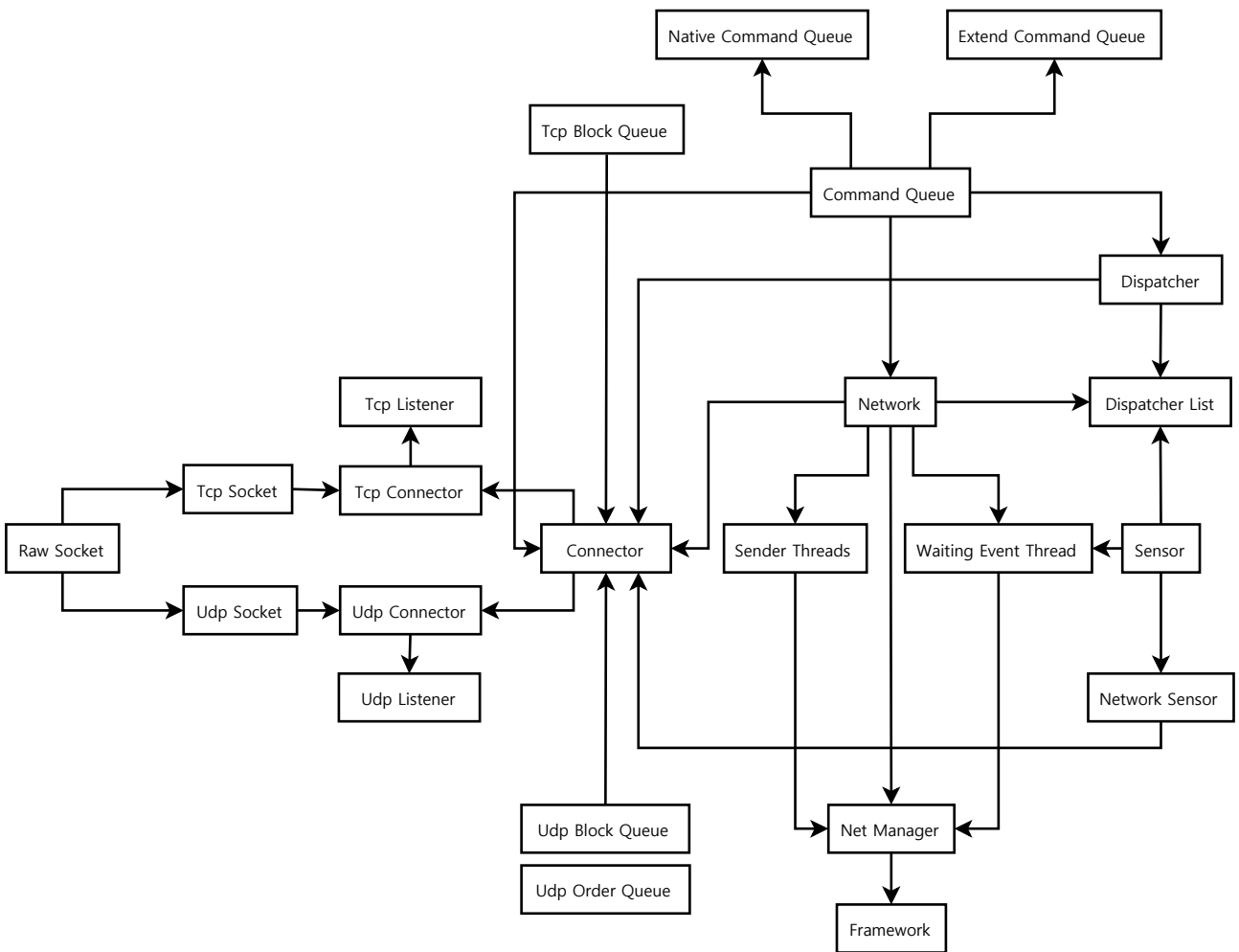
개발 방법이 동일하니 자신있는 언어로 개발 하세요.

확장 기능으로는 게임을 웹 서버로도 개발할 수 있도록 NDP over HTTP를 지원하고 있습니다. 프로토콜 규격만 준수 한다면 Node.js, Java/Spring, Asp.net 등 어떠한 Framework를 사용하여도 무방합니다.

물론 JSON, XML, HTML 등도 배려되어 사용할 수 있지만 게임 패킷으로는 가장 콤팩트 한 NDP over HTTP를 사용하세요.

트래픽 양을 극단적으로 줄여줍니다.

# 넷드론 엔진 제품 소개



## 네트워크 엔진 구조

## 시스템 요구 사항

실행 환경	Ubuntu 14.04 LTS 이상
	CentOS 6.7 이상
	Windows 7 이상
	OSX 10 이상
개발 도구	Unity Editor 5 이상
	Monodevelop 5 이상
	Visual Studio 2010 이상
	Eclipse Luna 이상

넷드론 엔진은 Windows, OSX, Linux, iOS, Android 플랫폼에서 테스트 되었습니다.

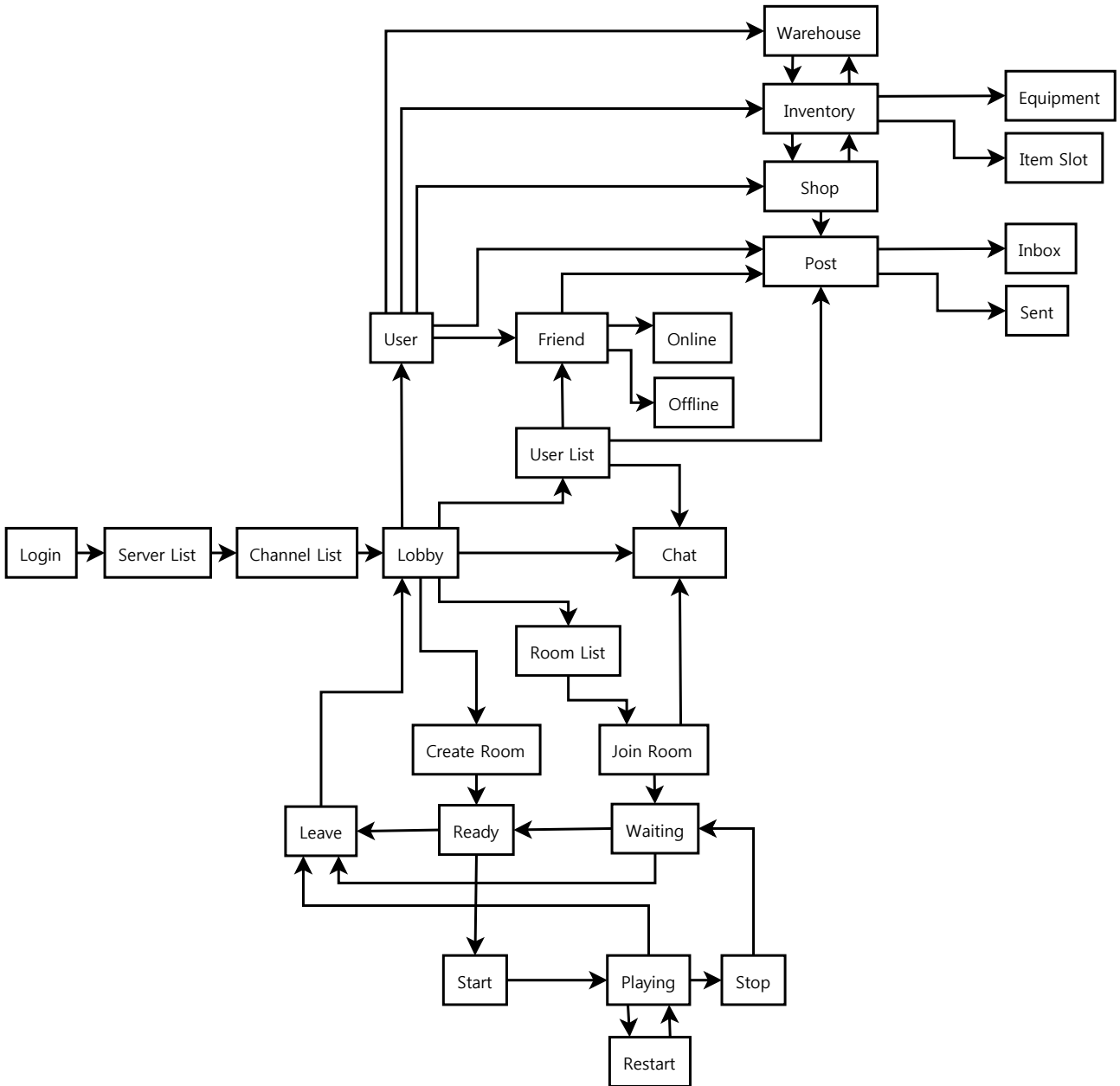
## 제품 구성

지원 기능	TCP, Reliable UDP	
	NDP, NDP over HTTP	
	Multi-Platform	Linux, Windows, Mac
	Multi-Language	C#, C++, PHP
	Multi-Threading	
	RC5/RC6 Packet Encryption/Decryption	
	Reuse memory allocation	
	Singleton Pattern	
	Factory Pattern	
	Compact Packet Design	TCP Header: 8 bytes UDP Header: 12 bytes
	Compact Game Log	
	Tracer	
	Poll(Overlapped IO), Epoll	
	Raw Sockets	
	CSV, Binary CSV	
Packet Queue		
예제	Room, Chat, Web examples for Unity3D	SmartClient
	C# based NDP server	StudyServer
	C++ based NDP server	UnityServer
	PHP based NDP over HTTP server	HyperServer
도구	Console Client	DummyClient
	RC5/RC6 encrypt tester	CryptTester
	Reuse memory tester	ReuseMemory
	CSV to Binary CSV tool	DataConverter

게임 네트워킹에 필요한 모든 것이 넷드론 엔진에 있습니다.

# 상호작용 게임 구조

## 멀티플레이어 게임의 흐름

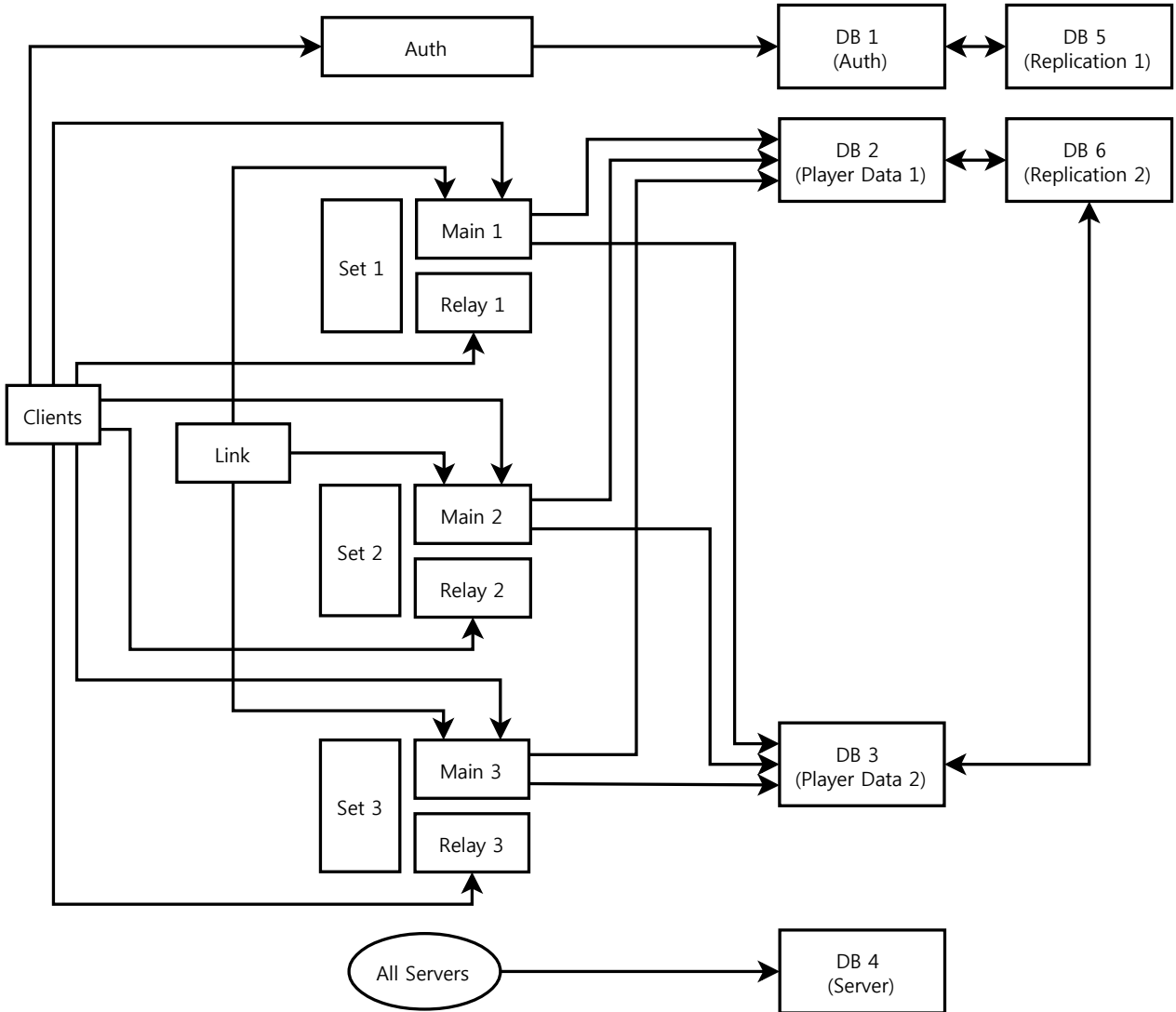


### 게임 시스템 구조

일반적인 MO(Multiplayer Online) 게임 시스템의 구조는 이러한 연결 흐름을 가지고 있습니다. 당신이 만약 MMO(Massively Multiplayer Online) 게임을 만들고 싶다면 Channel을 시스템 설계 단계에서 제거하세요.

MMO(Massively Multiplayer Online) 게임은 Broadcast 기준이 Channel이 아닌 Zone이 됩니다.

## 분산처리 서버 구성



### 서버 시스템 구조

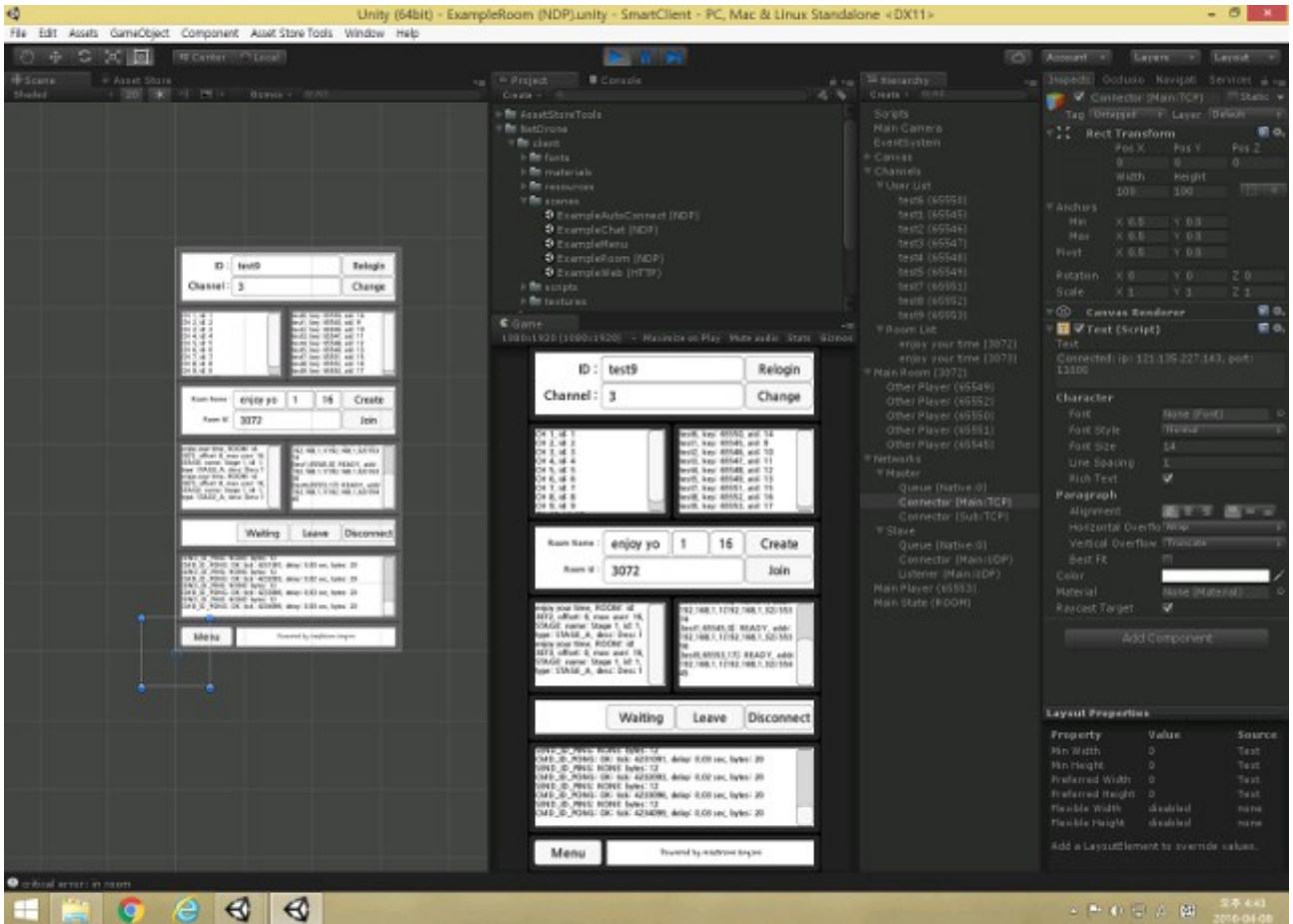
클라이언트가 서버에 로그인 하여 어떻게 다른 플레이어들과 통신 하는지와 서버간 통신에 대한 연결 흐름을 이해할 수 있는 순서입니다. 당신이 만약 멀티플레이어 게임을 만들고 싶다면 되도록 서버 시스템 구조를 참고하세요.

넷드론 엔진을 활용할 경우 Auth Server, Link Server, Relay Server, Main Server 등을 빠르고 안전하게 개발할 수 있습니다.

계획된 게임 시스템 설계에 맞게 서버간 연결 구조를 변경하세요.

# 개발 방법 알아보기

## Unity Editor로 C# 클라이언트 만들기



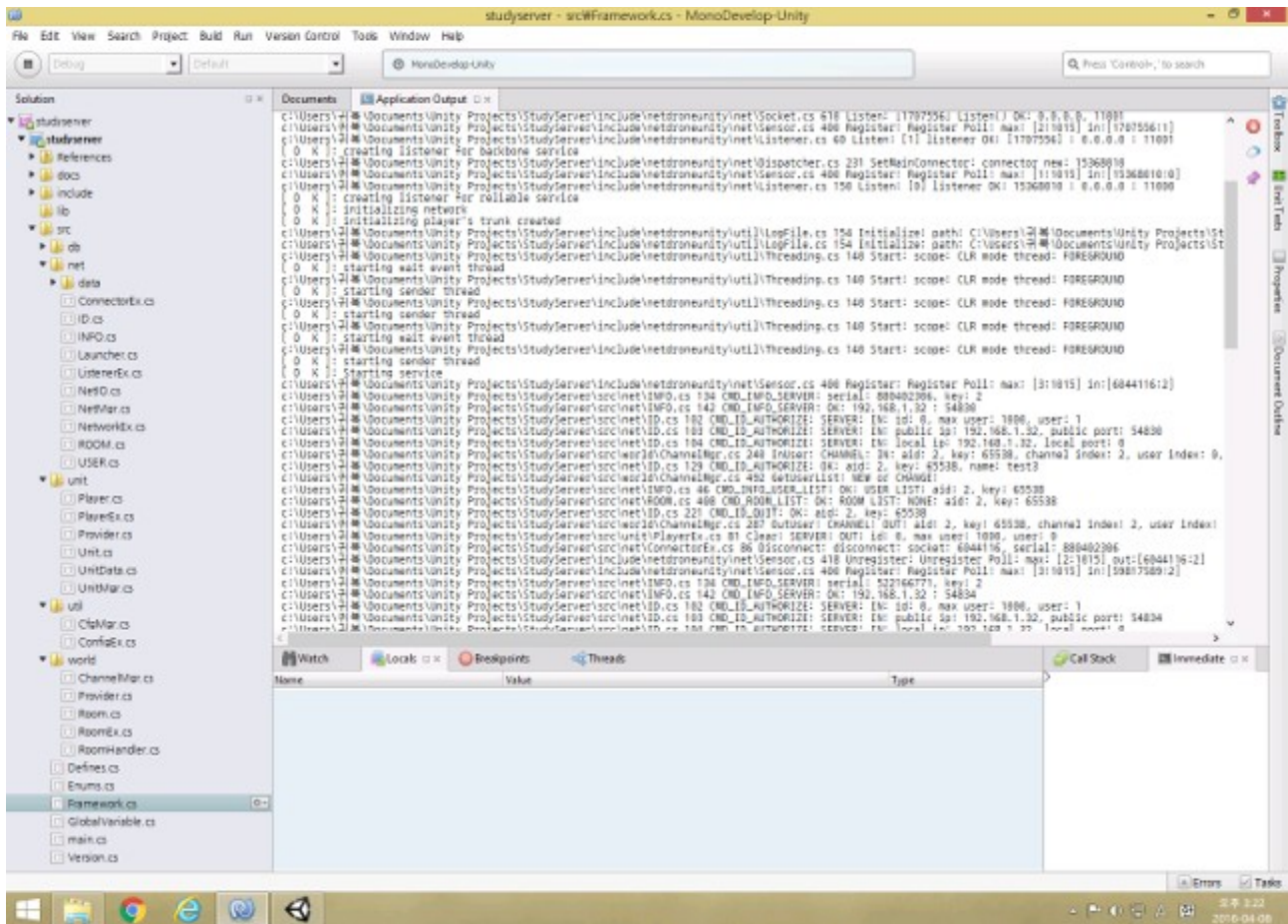
Unity IDE

넷드론 엔진의 클라이언트 예제는 패킷 흐름을 명확히 분석 할 수 있게 데이터로 모든것을 표현합니다. 게임 Scene으로는 볼 수 없었던 정확한 패킷 처리 단계 등을 예제를 통해 확인하세요.

대규모(복잡한) 게임을 제작할 경우 이런식으로 데이터 검증을 하게 됩니다.

이제 테스트용 Scene을 참고 하면서 게임 Scene 작업을 시작하세요.

# MonoDevelop으로 C# 독립형 서버 만들기



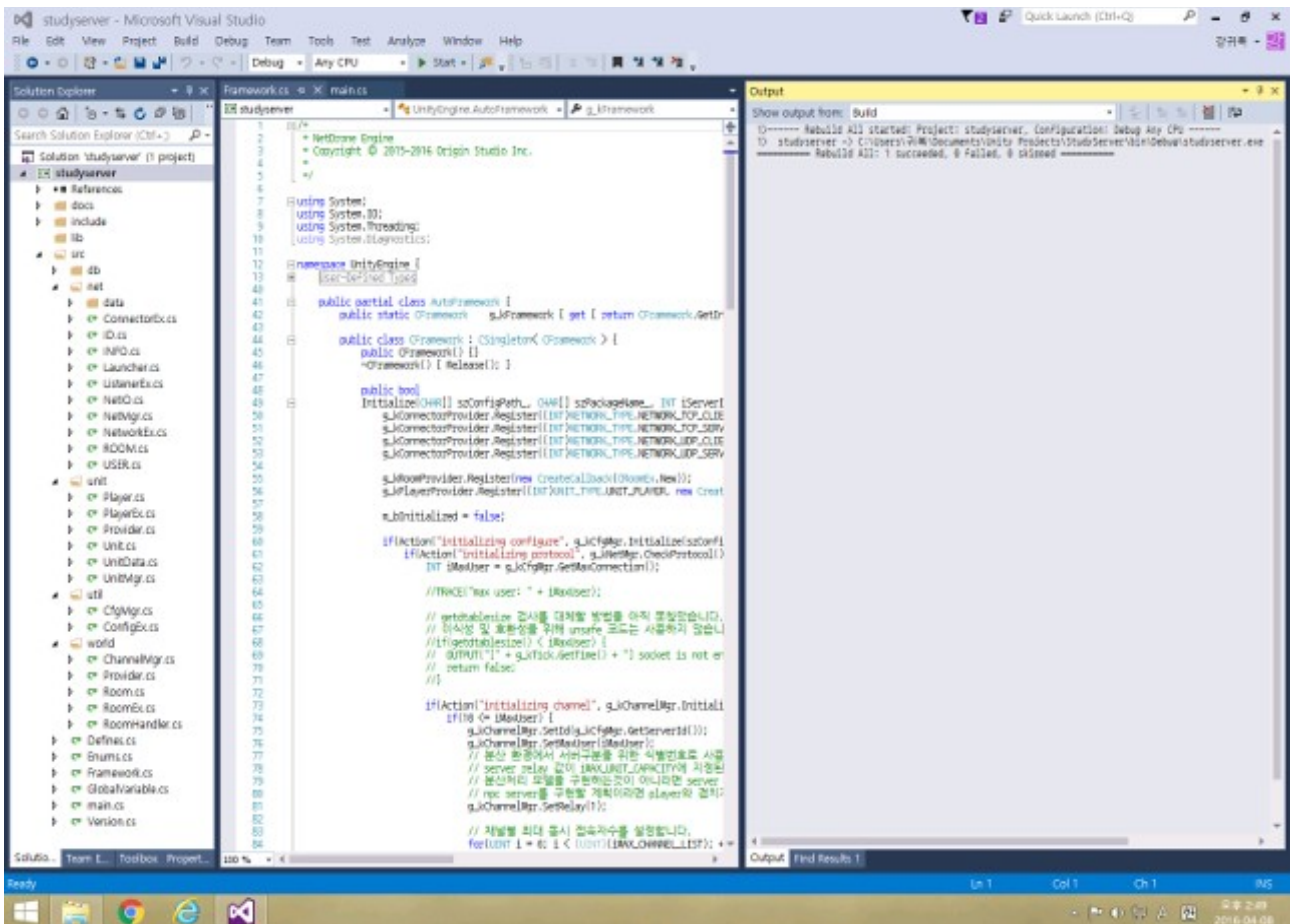
## MonoDevelop IDE

넷드론 유니티와 Study Server는 Mono / .NET 4.5 기반에서 제작 되었습니다. 테스트 된 실행 환경은 Windows, OSX, Linux 이며 MonoDevelop 5로 개발됩니다.

프로토콜 타입을 추가하면서 자신만의 게임 서버를 만들어 보세요.



# Visual Studio로 C# 독립형 서버 만들기

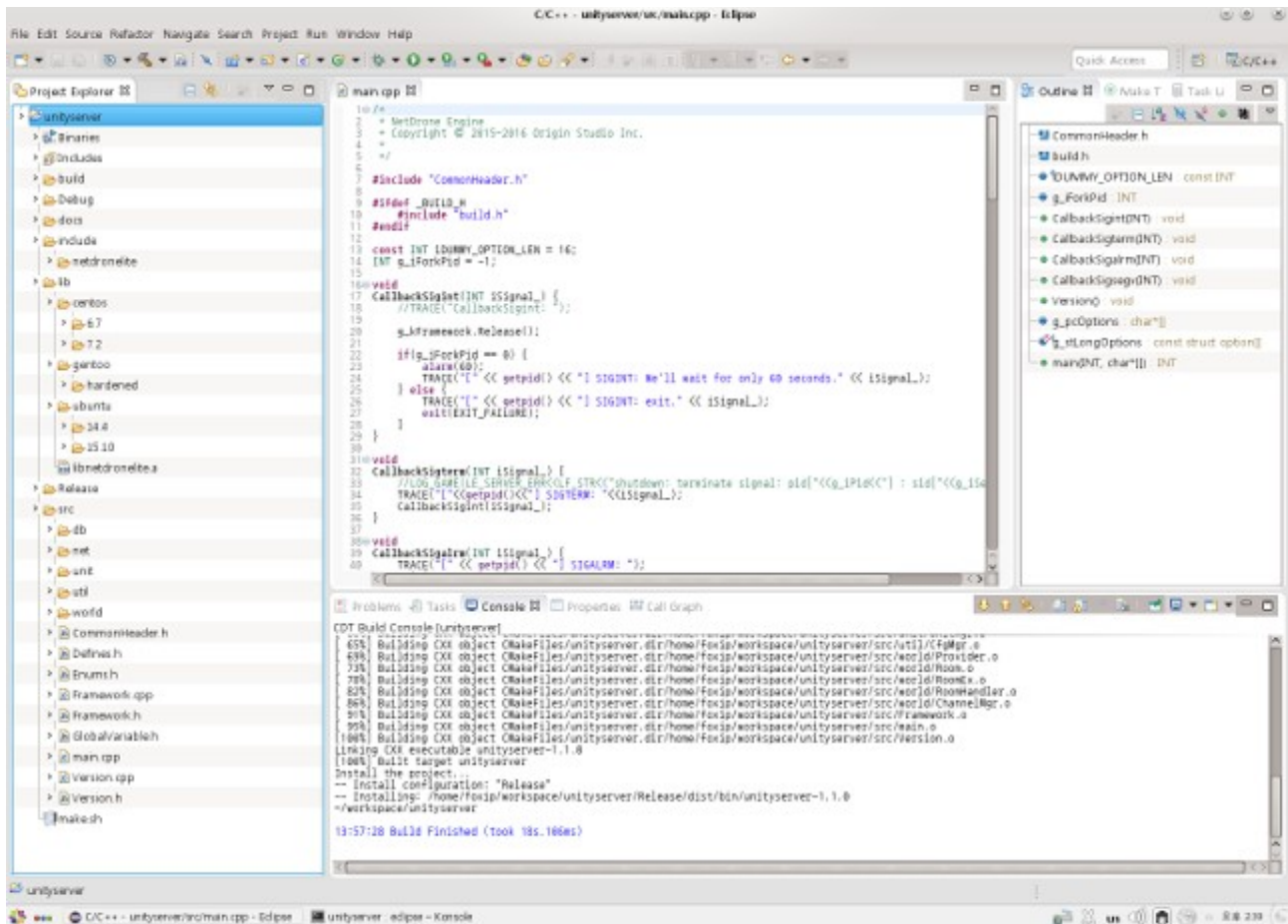


Visual Studio IDE

넷드론 유니티와 Study Server는 Windows의 Visual Studio 2010 / 2015와 .NET Framework 4.5를 지원합니다. Visual Studio 2010의 경우 프로젝트 구성을 다시해야 하지만 Visual Studio 2015의 경우 MonoDevelop 5 프로젝트와 호환됩니다.

프로토콜 타입을 추가하면서 자신만의 게임 서버를 만들어 보세요.

# Eclipse CDT로 C++ 독립형 서버 만들기

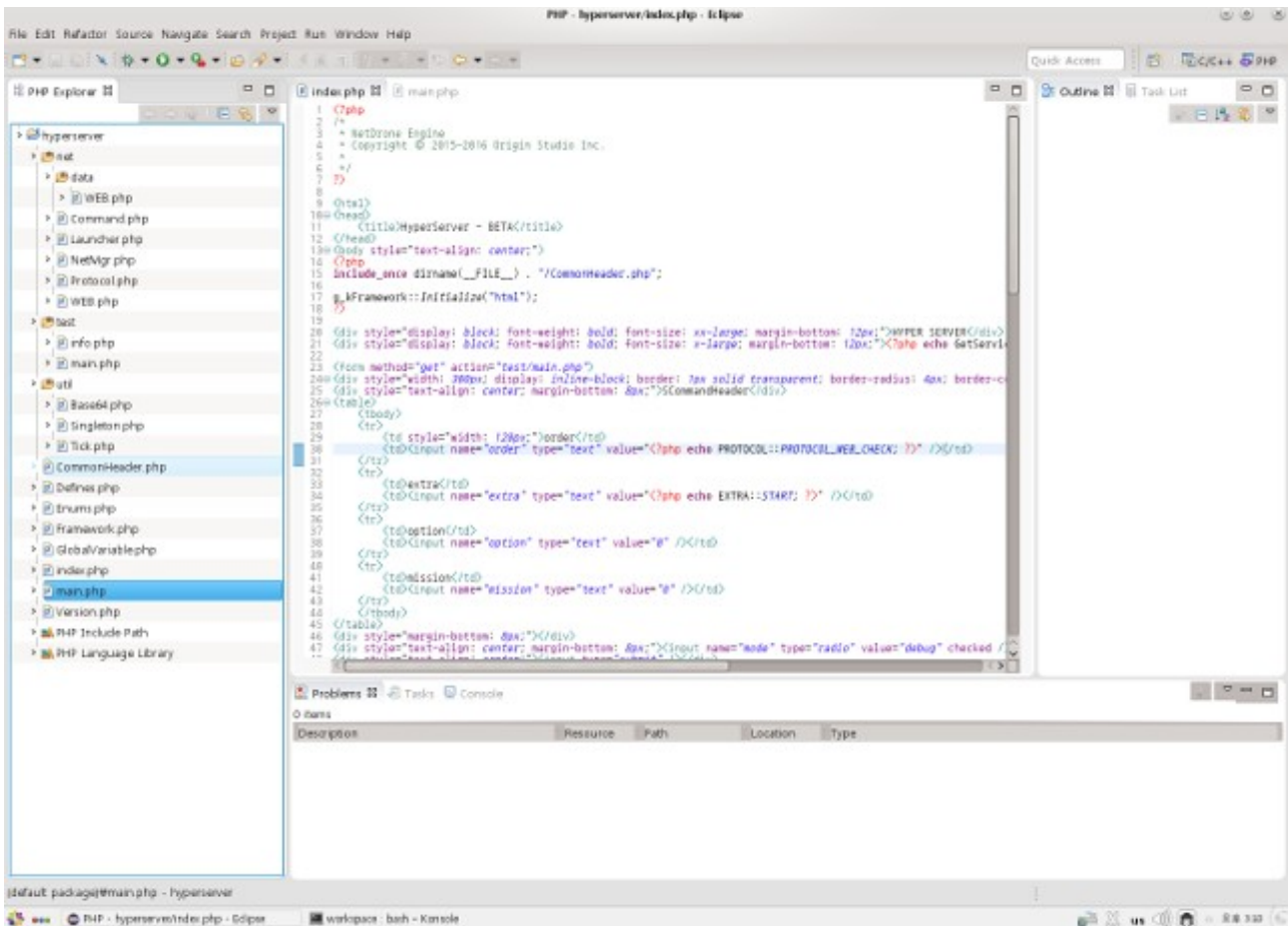


## Eclipse CDT

넷드론 라이트와 Unity Server는 Gentoo Linux의 GLIBC / GCC 환경에서 제작 되었습니다. 테스트 된 실행 환경은 Ubuntu 14 / 15, CentOS 6 / 7, Gentoo Hardened 이며 Eclipse CDT로 개발됩니다.

프로토콜 타입을 추가하면서 자신만의 게임 서버를 만들어 보세요.

# Eclipse PDT로 PHP 웹 서버 만들기



## Eclipse PDT

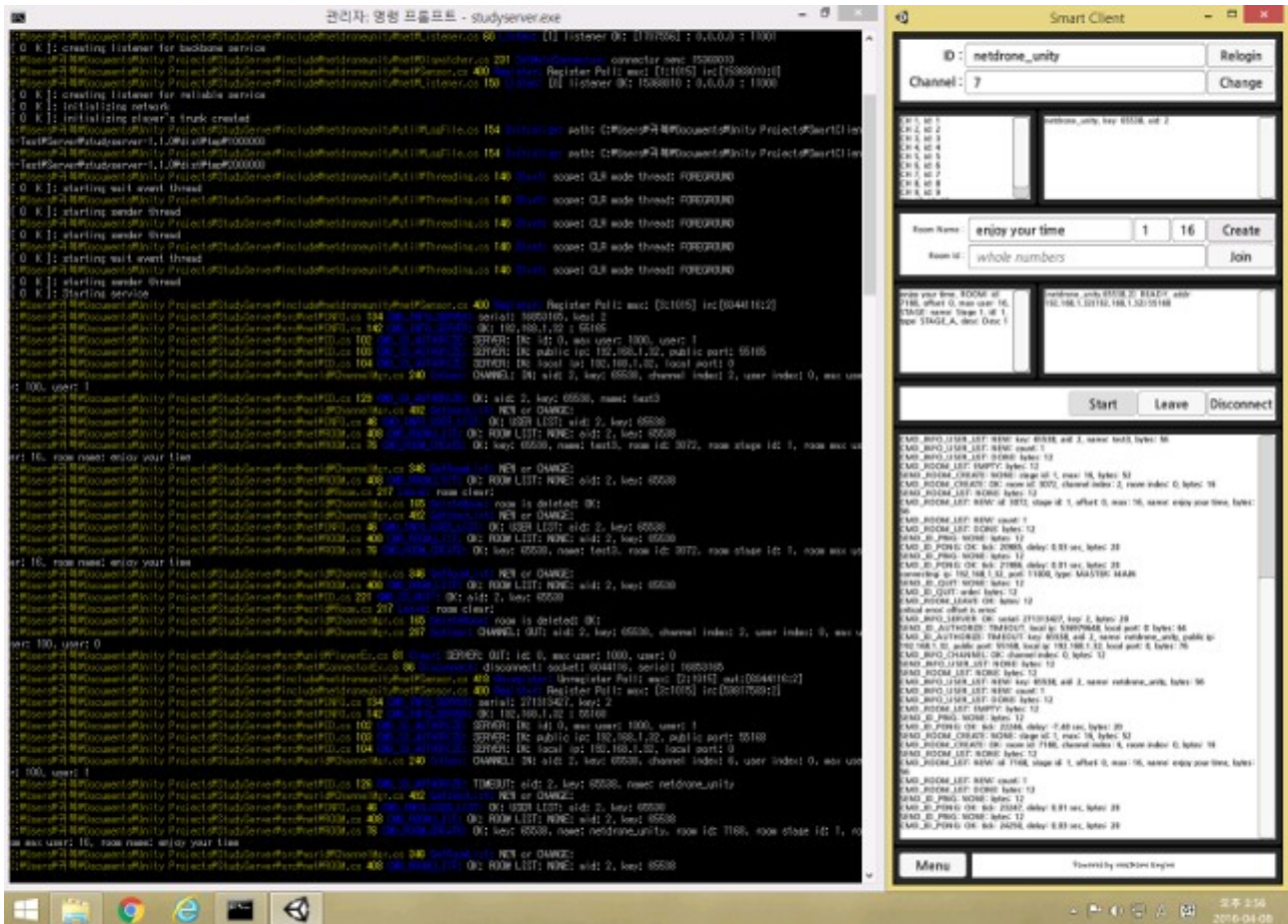
넷드론 엔진은 웹 서버로 구현이 가능한 SO(Single player Online) 게임을 위해 Hyper Server 예제를 제공합니다. PHP 5.3 환경에서 제작 되었으며 Eclipse PDT로 개발됩니다.

NDP over HTTP 규격을 사용하게 되면 기존 넷드론 기반의 클라이언트 기능을 모두 사용하는 것이 가능합니다.

프로토콜 타입을 추가하면서 자신만의 게임 서버를 만들어 보세요.

# 예제 및 도구 설명

## 서버 예제



### Study Server

당신이 만약 C# 프로그래머라면 Study Server 예제를 이용하세요. Unity Editor와 함께 편리한 개발이 가능합니다.

로컬에서 클라이언트를 여러개 실행하거나 팀원들이 내 주소로 접속 할 수 있습니다.

방화벽 설정도 잘 해주세요.



```

root@expressai:/home/data/games/unityserver
[root@expressai unityserver]# ls log
1000100-201604071130.log 1000100-201604080950.log 1000100-201604081038.log 1000100-201604081040.log 1000100-201604081140.log 1000100-201604081250.log 1000100-201604081310.log
1000100-201604071140.log 1000100-201604080950.log 1000100-201604081038.log 1000100-201604081040.log 1000100-201604081140.log 1000100-201604081250.log 1000100-201604081310.log
1000100-201604072150.log 1000100-201604080950.log 1000100-201604081038.log 1000100-201604081100.log 1000100-201604081100.log 1000100-201604081300.log 1000100-201604081540.log
[root@expressai unityserver]# cat log/1000100-201604081330.log
1460009947,6,7,9,13,tests3
1460009947,16,7,9,22,1072,23,1,4,enjoy your time
1460009940,7,7,9,13,tests3
1460009940,6,7,9,13,tests3
1460009950,16,7,9,22,1072,23,1,4,enjoy your time
1460009952,7,7,9,13,tests3
1460009062,6,7,9,13,tests3
1460009063,16,7,9,22,1072,23,1,4,enjoy your time
1460009063,16,7,9,22,1072,23,1,4,enjoy your time
1460009060,7,7,9,13,tests3
1460009062,6,7,9,13,tests3
1460009060,7,7,9,13,tests3
1460009060,6,7,9,13,tests3
[root@expressai unityserver]# tail -30 log/unityserver.log
/home/build/src/unityserver/src/serverId/ChannelMgr.cpp 224 OutServer CHANNEL: OUT: aid: 9, key: 65545, channel index: 1, user index: 0, max user: 100, user: 0
/home/build/src/unityserver/src/unit/PlayerTx.cpp 58 Clear SERVER: OUT: id: 100, max user: 1000, user: 0
/home/build/src/unityserver/src/net/ConnectToTx.cpp 53 Disconnect: disconnect socket: 9, serial: 302745720
/home/build/src/netfranelite/src/net/Sensor.cpp 357 Unregister: Unregister Epoll: max: [2:1015] out:[0:0]
/home/build/src/netfranelite/src/net/Sensor.cpp 326 Register: Register Epoll: max: [5:1015] in:[0:0]
/home/build/src/unityserver/src/net/INP.cpp 183 CND_INP_SERVER: serial: 1009593063, key: 9
/home/build/src/unityserver/src/net/INP.cpp 187 CND_ID_AUTH: OK: aid: 9, key: 65545
/home/build/src/unityserver/src/net/ID.cpp 74 CND_ID_AUTH: SERVER: IN: id: 100, max user: 1000, user: 1
/home/build/src/unityserver/src/net/ID.cpp 75 CND_ID_AUTH: SERVER: IN: public ip: 192.168.1.32, public port: 55007
/home/build/src/unityserver/src/net/ID.cpp 76 CND_ID_AUTH: SERVER: IN: local ip: 192.168.1.32, local port: 0
/home/build/src/unityserver/src/serverId/ChannelMgr.cpp 180 InServer CHANNEL: IN: aid: 9, key: 65545, channel index: 0, user index: 0, max user: 100, user: 1
/home/build/src/unityserver/src/net/ID.cpp 181 CND_ID_AUTH: OK: aid: 9, key: 65545, name: dummy
/home/build/src/unityserver/src/net/ID.cpp 187 CND_ID_AUTH: OK: aid: 9, key: 65545
/home/build/src/unityserver/src/unit/PlayerTx.cpp 58 Clear SERVER: OUT: id: 100, max user: 1000, user: 0
/home/build/src/unityserver/src/net/ChannelMgr.cpp 224 OutServer CHANNEL: OUT: aid: 9, key: 65545, channel index: 0, user index: 0, max user: 100, user: 0
/home/build/src/unityserver/src/net/ConnectToTx.cpp 53 Disconnect: disconnect socket: 9, serial: 3009593063
/home/build/src/netfranelite/src/net/Sensor.cpp 357 Unregister: Unregister Epoll: max: [2:1015] out:[0:0]
/home/build/src/netfranelite/src/net/Sensor.cpp 326 Register: Register Epoll: max: [3:1015] in:[0:0]
/home/build/src/unityserver/src/net/INP.cpp 183 CND_INP_SERVER: serial: 65003079, key: 9
/home/build/src/unityserver/src/net/INP.cpp 187 CND_ID_AUTH: OK: aid: 9, key: 65545
/home/build/src/unityserver/src/net/ID.cpp 74 CND_ID_AUTH: SERVER: IN: id: 100, max user: 1000, user: 1
/home/build/src/unityserver/src/net/ID.cpp 75 CND_ID_AUTH: SERVER: IN: public ip: 192.168.1.32, public port: 55000
/home/build/src/unityserver/src/net/ID.cpp 76 CND_ID_AUTH: SERVER: IN: local ip: 192.168.1.32, local port: 0
/home/build/src/unityserver/src/serverId/ChannelMgr.cpp 180 InServer CHANNEL: IN: aid: 9, key: 65545, channel index: 1, user index: 0, max user: 100, user: 1
/home/build/src/unityserver/src/net/ID.cpp 181 CND_ID_AUTH: OK: aid: 9, key: 65545, name: dummy_test
/home/build/src/unityserver/src/net/ID.cpp 187 CND_ID_AUTH: OK: aid: 9, key: 65545
/home/build/src/unityserver/src/serverId/ChannelMgr.cpp 224 OutServer CHANNEL: OUT: aid: 9, key: 65545, channel index: 1, user index: 0, max user: 100, user: 0
/home/build/src/unityserver/src/unit/PlayerTx.cpp 58 Clear SERVER: OUT: id: 100, max user: 1000, user: 0
/home/build/src/unityserver/src/net/ConnectToTx.cpp 53 Disconnect: disconnect socket: 9, serial: 65003079
/home/build/src/netfranelite/src/net/Sensor.cpp 357 Unregister: Unregister Epoll: max: [2:1015] out:[0:0]
[root@expressai unityserver]#

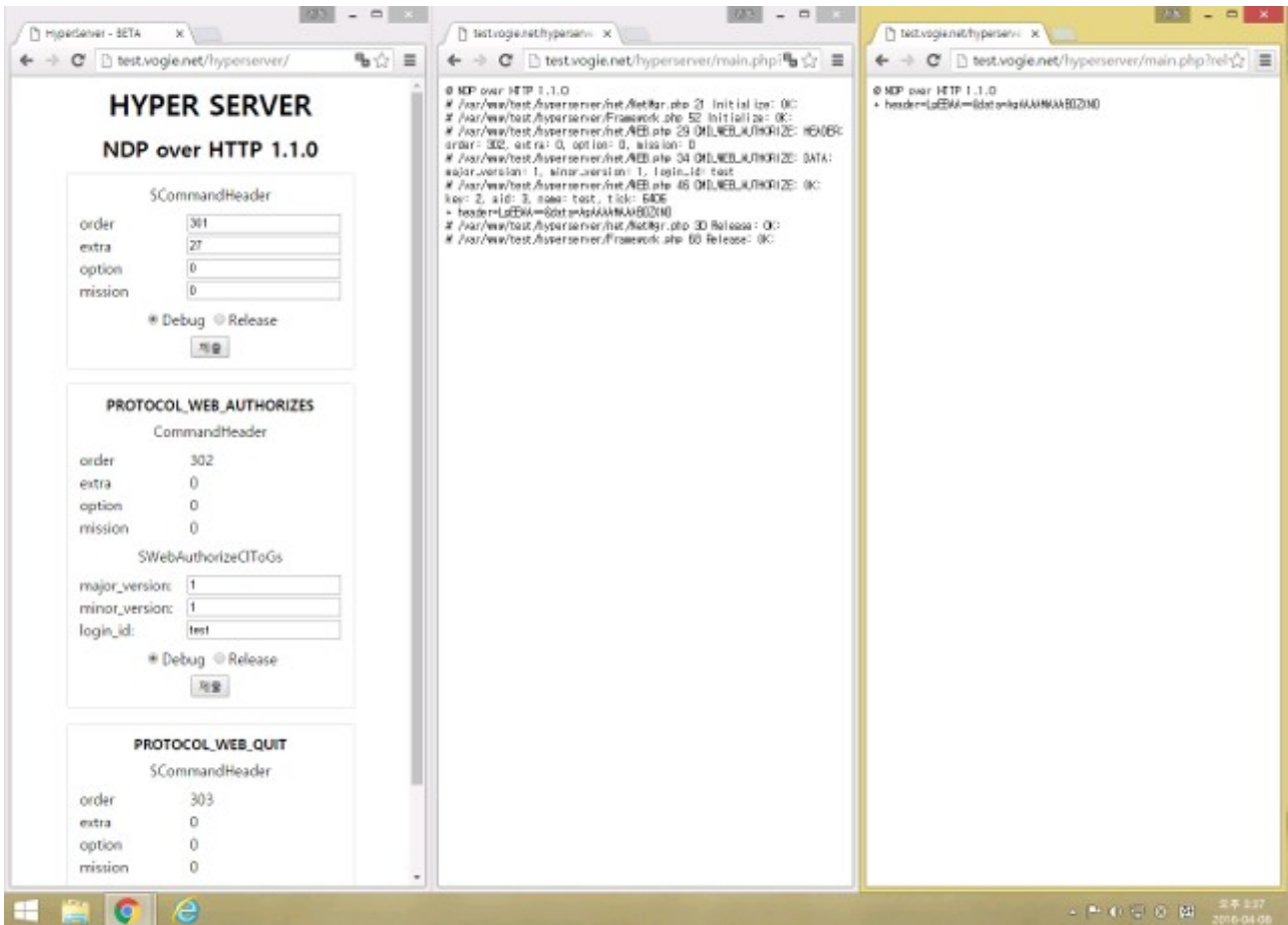
```

## Unity Server

당신이 만약 C++ 프로그래머라면 Unity Server 예제를 이용하세요. Native Code만의 강력한 성능을 경험할 수 있습니다.

VirtualBox 등을 이용해 Linux를 설치하면 로컬에서 개발하는 것이 가능하며 브리지 어댑터를 이용할 경우 팀원들이 내 주소로 접속 할 수 있습니다.

방화벽 설정도 잘 해주세요.



## Hyper Server

당신이 만약 PHP 프로그래머라면 Hyper Server 예제를 이용하세요. 익숙한 웹 서버로 개발이 가능합니다.

GET Query String으로 Debug를 명시할 경우 디버깅용 로그가 포함 됩니다.

프로토콜 테스트 폼으로 확인하세요.

## 클라이언트 예제

```
관리자: 명령 프롬프트 - DummyClient.exe
input remote ip (default: 127.0.0.1): 192.168.1.11
input remote port (default: 11000):
input connector type (default: tcp, 1: tcp, 2: reliable udp): 1
input header crypt (default: true, 1: false, 2: true):
input login id (default: dummy): dummy_test
input channel id (default: 1): 2

[ D : X ]: Initializing network
[ D : X ]: creating connector for service

MAX_PACKET_SIZE: 1440
TCP_PACKET_HEAD_SIZE: 8
UDP_PACKET_HEAD_SIZE: 12
COMMAND_HEAD_SIZE: 4
COMMAND_DATA_SIZE: 1424
PACKET_DATA_SIZE: 1428
SIdAuthorizeToCI: 84
SIdAuthorizeCIToCs: 52
SIdPongToCI: 8
SIdInfoServerToCI: 8

remote ip: 192.168.1.11, remote port: 11000
connector type: tcp
header crypt: true
channel index: 1
login id: dummy_test

netdroneunity 1.1.0 (2016-04-08 오후 1:38:22)
DummyClient 1.1.0 (2016-04-08 오후 1:39:22)
non-thread

Login: connector type: tcp: connected
CMD_INFO_SERVER: OK: serial: 86088879, key: 9, bytes: 20
Update: update: check: tick: 49, current: 0
SEND_ID_PING: NONE: bytes: 12
SEND_ID_AUTHORIZE: NONE, local ip: 198979649, local port: 0, bytes: 84
CMD_ID_AUTHORIZE: OK: key: 65545, eid: 9, name: dummy_test, public ip: 192.160.1.32, public port: 55000, local ip: 192.160.1.32, local port: 0, bytes: 76
CMD_INFO_CHANNEL: CHECK: order: 108, order: 1, option: 0, mission: 0, bytes: 12
Update: update: check: tick: 3875552, current: 3875518
SEND_ID_PING: NONE: bytes: 12
CMD_ID_PING: OK: tick: 3875526, delay: 0.00 sec, bytes: 20
Update: update: check: tick: 3875557, current: 3875522
SEND_ID_PING: NONE: bytes: 12
CMD_ID_PING: OK: tick: 3875590, delay: 0.05 sec, bytes: 20
Update: update: check: tick: 3877562, current: 3877527
SEND_ID_PING: NONE: bytes: 12
CMD_ID_PING: OK: tick: 3877595, delay: 0.05 sec, bytes: 20
```

### Dummy Client

넷드론 유니티는 유니티 엔진 없이도 독립적인 사용이 가능합니다. Dummy Client 예제는 TCP와 Reliable UDP를 구현하고 있으며 넷드론 기반 클라이언트 구조를 알기 쉽게 만든 프로그램 입니다.

Login 후 연결 유지를 위해 Ping / Pong 패킷을 전송합니다.

Smart Client를 실행 해보면 Dummy Client를 찾을 수 있어요.





```
관리자: 명령 프롬프트 - ReuseMemory.exe
C:\Users\기\Documents\Unity Projects\SmartClient-Test\Tools\ReuseMemory#dist>ReuseMemory.exe
input allocation count (default: 8):
input additional count (default: 3):
input retry count (default: 3):

allocation count: 3
additional count: 8
retry count: 3

new: instance: 21088178
new: instance: 55530882
new: instance: 30015890

renew: instance: 30015890
renew: instance: 55530882
renew: instance: 21088178
new: instance: 1707556
new: instance: 15388010
new: instance: 4034363

renew: instance: 4034363
renew: instance: 15388010
renew: instance: 1707556
renew: instance: 21088178
renew: instance: 55530882
renew: instance: 30015890
new: instance: 38848274
new: instance: 63208015
new: instance: 82001227

free: count: 0, instance: 82001227, next: 63208015
free: count: 1, instance: 63208015, next: 38848274
free: count: 2, instance: 38848274, next: 30015890
free: count: 3, instance: 30015890, next: 55530882
free: count: 4, instance: 55530882, next: 21088178
free: count: 5, instance: 21088178, next: 1707556
free: count: 6, instance: 1707556, next: 15388010
free: count: 7, instance: 15388010, next: 4034363
free: count: 8, instance: 4034363, next: null
press any key to exit.
```

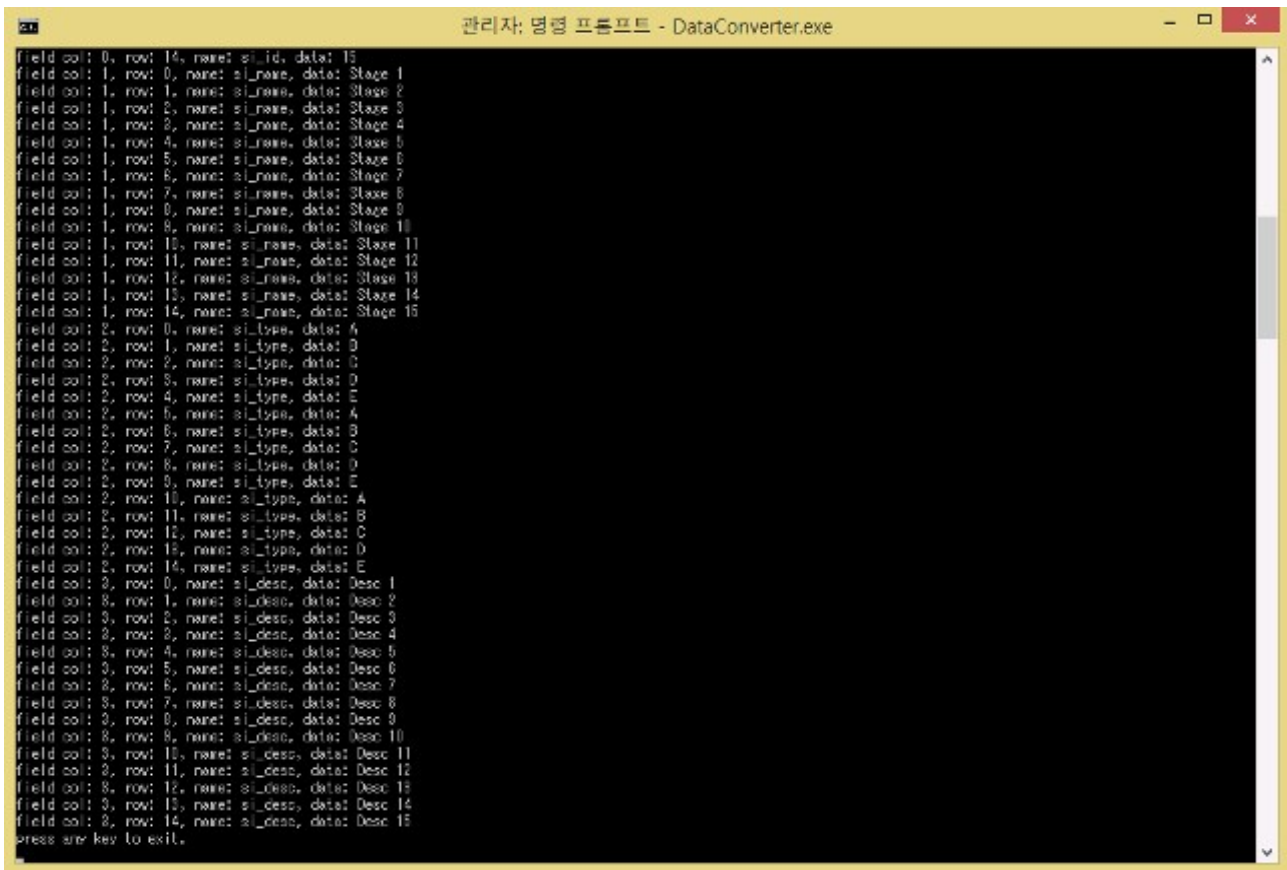
## Reuse Memory

넷드론 엔진은 메모리 재사용 및 할당 속도 개선을 위해 New와 Delete를 구현하였습니다.

CAllocator를 상속한 모든 클래스는 메모리 재사용이 가능합니다.

Reuse Memory 예제를 실행하여 패킷 생성 및 재사용을 실험하세요.

## CSV to Binary CSV



```
관리자: 명령 프롬프트 - DataConverter.exe
field col: 0, row: 14, name: si_id, data: 15
field col: 1, row: 0, name: si_name, data: Stage 1
field col: 1, row: 1, name: si_name, data: Stage 2
field col: 1, row: 2, name: si_name, data: Stage 3
field col: 1, row: 3, name: si_name, data: Stage 4
field col: 1, row: 4, name: si_name, data: Stage 5
field col: 1, row: 5, name: si_name, data: Stage 6
field col: 1, row: 6, name: si_name, data: Stage 7
field col: 1, row: 7, name: si_name, data: Stage 8
field col: 1, row: 8, name: si_name, data: Stage 9
field col: 1, row: 9, name: si_name, data: Stage 10
field col: 1, row: 10, name: si_name, data: Stage 11
field col: 1, row: 11, name: si_name, data: Stage 12
field col: 1, row: 12, name: si_name, data: Stage 13
field col: 1, row: 13, name: si_name, data: Stage 14
field col: 1, row: 14, name: si_name, data: Stage 15
field col: 2, row: 0, name: si_type, data: A
field col: 2, row: 1, name: si_type, data: B
field col: 2, row: 2, name: si_type, data: C
field col: 2, row: 3, name: si_type, data: D
field col: 2, row: 4, name: si_type, data: E
field col: 2, row: 5, name: si_type, data: A
field col: 2, row: 6, name: si_type, data: B
field col: 2, row: 7, name: si_type, data: C
field col: 2, row: 8, name: si_type, data: D
field col: 2, row: 9, name: si_type, data: E
field col: 2, row: 10, name: si_type, data: A
field col: 2, row: 11, name: si_type, data: B
field col: 2, row: 12, name: si_type, data: C
field col: 2, row: 13, name: si_type, data: D
field col: 2, row: 14, name: si_type, data: E
field col: 3, row: 0, name: si_desc, data: Desc 1
field col: 3, row: 1, name: si_desc, data: Desc 2
field col: 3, row: 2, name: si_desc, data: Desc 3
field col: 3, row: 3, name: si_desc, data: Desc 4
field col: 3, row: 4, name: si_desc, data: Desc 5
field col: 3, row: 5, name: si_desc, data: Desc 6
field col: 3, row: 6, name: si_desc, data: Desc 7
field col: 3, row: 7, name: si_desc, data: Desc 8
field col: 3, row: 8, name: si_desc, data: Desc 9
field col: 3, row: 9, name: si_desc, data: Desc 10
field col: 3, row: 10, name: si_desc, data: Desc 11
field col: 3, row: 11, name: si_desc, data: Desc 12
field col: 3, row: 12, name: si_desc, data: Desc 13
field col: 3, row: 13, name: si_desc, data: Desc 14
field col: 3, row: 14, name: si_desc, data: Desc 15
press any key to exit.
```

### Data Converter

넷드론 유니티는 CSV 및 Binary CSV Loader를 제공합니다. 또한 Data Converter 도구를 이용하여 CSV to Binary CSV로 변환할수 있습니다.

Binary CSV는 많은 양의 데이터 처리에 유리하지만 잘못 입력된 게임 데이터 확인이 불편합니다.

개발할 때는 CSV를 배포할때는 Binary CSV를 사용해보세요.

# 구입 안내

## 인디 & 스튜디오

에셋스토어 구입 비용 \$50가 전부입니다. 넷드론 유니티는 엔진 소스가 제공되며 동시접속 수에 제한이 없는 상용화 가능 솔루션입니다.

또한 구매 여부와 상관없이 포럼을 통해 온라인 기술지원을 해드립니다.

에셋 스토어: <http://u3d.as/mzz>

포럼: <http://www.vogie.net>

## 기업

넷드론 엔터프라이즈는 분산처리 서버 및 DB 예제가 제공되며 넷드론 라이트의 엔진 소스가 포함 됩니다.

또한 협의 내용에 따라 교육 및 컨설팅, 커스터마이징, 오프라인 기술지원 등이 가능하며 상용화를 위한 집중을 해드립니다.

### 넷드론 엔터프라이즈 기본 구성

- 기업용 엔진 영구 라이선스 (소스코드 포함)
- 패킷 암호화 확장 기능
- 분산 처리 서버 및 RDBMS 예제 (MySQL)
- 무료 기술지원 (1년)

### 제품 가격

- 넷드론 엔터프라이즈: 20,000,000원 부터
- 주문제작: 100,000,000원 부터

정확한 가격은 해당 프로젝트의 요청 사항 및 기간 등을 기준으로 산출합니다. 연락처로 문의 사항을 보내주시면 신속히 답변을 드리겠습니다.

필요한 자료 요청이나 질문 등을 보내주세요.

감사합니다.

이메일: [contact@vogie.net](mailto:contact@vogie.net)

포럼: <http://www.vogie.net>