**GameServer**

Affinity.cpp - affinity related content class

AlternateMerchant.cpp - Disabled

Apet.cpp - Pet related content offensive class

APetProto.cpp - attacking pet-related DB data class

Area.cpp - Area-related classes

Assist.cpp - buff / debuff related classes

BasePoolManager.cpp - Memory pool class

Battle.cpp - combat-related processing functions

Battle\_DivisionExpSP.cpp - After the battle experience / skill division processing functions

Battle\_GetDamage.cpp - Damage to obtain combat function

Battle\_IsAvailable.cpp - a function that checks whether the battle

Battle\_NPC\_DropItem.cpp - After the battle NPC function to drop the item

Battle\_ProcAfterHit.cpp - Attack Damage is processed after the applicable

Battle\_ProcDead\_APet.cpp - attacking pets this function to handle when he died after a battle

Battle\_ProDead\_Elemental.cpp - function to handle when your pet dies after battle

Battle\_ProcDead\_NPC.cpp - a function that NPC to handle when he died after a battle

Battle\_ProcDead\_PC.cpp - function PC to handle when he died after a battle

Battle\_ProcDead\_Pet.cpp - a function that handles ttyae pet dies after battle

Cell.cpp - cell related classes

ChannelSettingManager - channel-specific content management classes

Character.cpp - Character Classes

Cmd.cpp - message command class

CmdMsg.cpp - Message Configuration function bar

ConfigFile.cpp - newStobm.bin file management class

Craft.cpp - gathering the relevant class (not used)

CraftManager.cpp - gathering relevant management class (not used)

CraftPoint.cpp - gathering point (not used)

DBCmd.cpp - DB connection and query input and output management classes

DBThread.cpp - a class that is used when the character storage

Decompose.cpp - decomposition of related classes (not used)

DecomposeManager.cpp - degradation associated management class (not used)

Descriptor.cpp - the socket file descriptor class

doFuncAction.cpp - action-specific message handling functions

doFuncAdmin.cpp - operator command message processing function

doFuncAffinity.cpp - affinity related message handling functions

doFuncalternateMerchant.cpp - (not used)

doFuncAttack.cpp - an attack message processing function

doFuncCashItem.cpp - Cash items related to message processing function

doFuncChangeJob.cpp - Ex-specific message handling functions

doFuncChat.cpp - Chat related messaging function

doFuncCraft.cpp - gathering relevant messaging function

doFuncDecompose.cpp - degradation associated message handling functions

doFuncEtc.cpp - so (-\_- ;;;) message handling functions

doFuncEvent.cpp - Event message handling functions

doFuncEventAutomation.cpp - (not used)

doFuncExAPet.cpp - attacking pets messaging function

doFuncExCastleMap.cpp - Siege map message handling functions

doFuncExCastllan.cpp - lords Costume relevant messaging function

doFuncExchange.cpp - exchange-related messaging function

doFuncExPartyMatch.cpp - party matching system processing functions

doFuncExpedition.cpp - Expedition messaging function

doFuncExPet.cpp - Pet related message handling functions

doFuncExSocket.cpp - equipment item socket-related messaging function

doFuncExSubJob.cpp - Kami professional traders message processing function

doFuncExtremeCube.cpp - Cube-related messaging function

doFuncFactory.cpp - making related to message processing function

doFuncFriend.cpp - friend relevant messaging function

doFuncGuild.cpp - Guild relevant messaging function

doFuncItem.cpp - items related to message processing function

doFuncLoot.cpp - (not used)

doFuncMemPos.cpp - Memory Scroll relevant messaging function

doFuncMonsterCombo.cpp - Monster Combo relevant messaging function

doFuncMove.cpp - Move-related messaging function

doFuncNPCRegen.cpp - Single dungeon NPC regeneration messaging function

doFuncParty.cpp - related party messaging function

doFuncPersonalShop.cpp - individual store-specific message handling functions

doFuncPK.cpp - PK-related messaging function

doFuncProduce.cpp - production-related messaging function

doFuncQuest.cpp - Quest messaging function

doFuncQuickSlot.cpp Quickslot messaging function

doFuncRaid.cpp - Raid Dungeon messaging function (entry, exit, etc.)

doFuncRebirth.cpp - Resurrection messaging function

doFuncReformSystem.cpp - original items reform-related message handling functions

doFuncSkill.cpp - skills related message handling functions

doFuncSSkill.cpp - special skills related to message processing function

doFuncStash.cpp - Warehouse-related messaging functiondoFuncTeach.cpp- The Guardian / Apprentice related message handling functions

doFuncTradeAgent.cpp - Stat deal messaging function

doFuncTrigerEvent.cpp - Dungeon Raid trigger event messaging function

doFuncWarp.cpp - Warp-related messaging function

DratanCastle.cpp - Tan siege drama class

DropItem.cpp - Event Automation DropItem processing class

Elemental.cpp - pet class

Env.cpp - (not used)

eventAutomation.cpp - Event Automation class

ExchangeItems.cpp - Item exchange class

Exp.cpp - Experience table

Expedtion.cpp - Expedition Class

ExtremeCube.cpp - Extreme Cube class

Factory.cpp - making classes

Fortune\_system.cpp - Costume Fortune 2 System class

Friend.cpp - Friends Management Class

GameServer.cpp - main function

GMCmd.cpp - operator command management classes

GMCmdList.cpp - Operator Command List class

Guild.cpp - Guild Class

GuildKick.cpp - Guild Master automatic expulsion class

hardcoding.cpp - many hard-coded

InputBuffer.cpp - reads the message stored in the input buffer Descriptor

IntergrationServer.cpp - integrated server class (villa, in the Kami and European servers)

Inventory.cpp - Inventory class

Item.cpp - Item Class

JumpingSystem.cpp - jumping system classes (level jump)

LacaRette.cpp - let Rakhine (not used)

localize.cpp - (not used)

Log.cpp - log

loot.cpp - (not used)

LootManager.cpp - (not used)

LuckyDrawBox.cpp - Lucky Draw Box (having the box kkamyeon items)

MapAttr.cpp - Map Properties class

MemPos.cpp - Memory Scroll class

MobActivity.cpp - NPC AI function

MonsterCombo.cpp - Monster Combo class

MonsterMercenary.cpp - Monster mercenary class

NPC.cpp - NPC class

NPCProto.cpp - NPC DB data

NPCRegenInfo.cpp - NPC Respawn information class

nProtectFunc.cpp - nProtect API

Option.cpp - Item Option Classes

OutputBuffer.cpp - Message output buffer

Party.cpp - party class

Party\_recall\_manager.cpp - party recall management class (singleton)

PC.cpp - PC class

PCList.cpp - PC list class

PersonalShop.cpp - Personal Store class

Pet.cpp - Pet class

PostOffice.cpp - postal system class

PostOfficeAgent.cpp - postal system class

PostOfficeAttachmentBase.cpp - postal system class

ProcConnMsg.cpp - message processing received from the connector

ProcHelper.Msg.cpp - message received from the helper handle

ProcMsgrMsg.cpp - message processing received from the Messenger

ProcSubHelperMsg.cpp - message processing received from the sub-helper

Product.cpp - production class

ProductManager.cpp - production management classes

ProgressTime.cpp - Disabled

Promotion2Event.cpp - Promo 2 class

PuzzleGame.cpp - Christmas 2012 puzzle class

Quest.cpp - Quest class

RaidData.cpp - Dungeon Raid Expedition

Random.cpp - random function

RareOptionProto.cpp - a rare option management classes

Requital.cpp - Compensation Events

Reward.cpp - Reward System

RewardEventItemManager.cpp - compensation class attendance system

RockPaperScissorsGame.cpp - Christmas 2012 Rock Paper Scissors game

RockPaperScissorsInfo.cpp - Christmas 2012 Janken information

RockPaperScissorsProb.cpp - Rock Paper Scissors odds, and Christmas 2012

Server.cpp - the main server class

ServerRun.cpp - the main loop

SetItemProto.cpp - Item Set DB data

Skill.cpp - Skills Class

SocketSystem.cpp - socket system class

SSkill.cpp - Special Skills Class

Stash.cpp - storage class

TitleSystem.cpp - Designation Class

TriggerSet.cpp - Dungeon Raid trigger settings

Utils.cpp - Utility Functions

WarCastle.cpp - Siege class

WarCastle\_Merac.cpp - Beta Ursae Majoris Siege

WarCastleFunc.cpp - Siege function

WarGround.cpp - Royal Rumble class

Zone.cpp - zone class

Connector

BillCmdMsg.cpp - billing message function bar

BillingClient.cpp - billing clients

BInputBuffer.cpp - Billings input buffer

BOutBuffer.cpp - Billing Output Buffer

BPacket.cpp - billing message (CNetMsg)

CheckPassword.cpp - United password verification class

CmdMsg.cpp - message function bar

ConfigFile.cpp - server configuration file class

Connector.cpp - connector server main function

DBCmd.cpp - DB connection class

Descriptor.cpp - socket descriptor class

InputBuffer.cpp - the input buffer class

localize.cpp - (not used)

Log.cpp - log file

md5.cpp - md5 function

OutputBuffer.cpp - the output buffer class

ProcConnMsg.cpp - message processing that comes with Connector

PromotionItem.cpp - Disabled

Random.cpp - random function

Server.cpp - Server Main class

ServerRun.cpp - Server Main Loop

User.cpp - user class

Utils.cpp - Utility Functions

WaitPlayerList.cpp - waiting Class

Helper

CmdInterpreter.cpp - Message to distinguish who is a helper

CmdMsg.cpp - message function

ConfigFile.cpp - server configuration file class

DBCmd.cpp - DB connection class

Descriptor.cpp - socket descriptor class

doFunc.cpp - received message handling functions

doFuncCHangeBoss.cpp - received message processing functions. Change Guild Master

doFuncCommand.cpp - received message handling functions

Expedition.cpp - Expedition Class

Guild.cpp - Guild Class

GuildKick.cpp - Guild Master automatic expulsion class

GuildStash.cpp - Guild storage class

Helper.cpp - Home Helpers

InputBuffer.cpp - the input buffer

Log.cpp - log

OutputBuffer.cpp - the output buffer

Party.cpp - party class

Random.cpp - random function

Server.cpp - the main server helper class

ServerRun.cpp - the main loop

TradeAgent.cpp - Trading agent class

Utils.cpp - Utilities

LoginServer

CheckPassword.cpp - US Login Authentication

CmdMsg.cpp - message function

ConfigFile.cpp - server configuration file class

DBCmd.cpp - DB connection class

Descriptor.cpp - socket descriptor class

InputBuffer.cpp - the input buffer

IntergrationServer.cpp - Integration Server (Villa, Kami and European)

IPCheck.cpp - US IP resolution table

Log.cpp - log

LoginServer.cpp - the main function

md5.cpp - md5 function

OutputBuffer.cpp - the output buffer

Rnadom.cpp - random function

Server.cpp - Server Main class

ServerRun.cpp - ropes

Utils.cpp function U team

WaitPlayerList.cpp - waiting list

Messenger

Cmd.cpp - message processing function receives the messenger

CmdMsg.cpp - message function

ConfigFile.cpp - server configuration file class

DBCmd.cpp - DB connection class

Descriptor.cpp - socket descriptor class

doFunc.cpp - received message handling functions

InputBuffer.cpp - the input buffer

Log.cpp - log

Messenger.cpp - the main function

OutputBuffer.cpp - the output buffer

Random.cpp - random function

Server.cpp - Server Main class

ServerRun.cpp - the main loop

Utils.cpp - Utility Functions

SubHelper

BackupMail.cpp - postal system backup

BasePoolManager.cpp - Pool Manager

CmdInterpreter.cpp - nine minutes message function

CmdMsg.cpp - message function

CommonQuery.cpp - the query string collection

ConfigFile.cpp - server configuration file class

DBCmd.cpp - DB connection class

Delivery.cpp - mail delivery system

Descriptor.cpp - socket descriptor class

doFunc.cpp - message handling functions

doFuncCommand.cpp - message handling functions

doFuncMail.cpp - mail processing functions

doFuncNotice.cpp - known processing functions

InputBuffer.cpp - the input buffer

LakaThread.cpp - thread manager

Log.cpp - log

MailLoader.cpp - email loader (not used)

NoticeSystemAgent.cpp - known systems management classes

OutputBuffer.cpp - the output buffer

pool.cpp - Disabled

PostOffice.cpp - post office mail system

PostOfficeAgent.cpp - post office mail system agents

PostOfficeAttachmentBase.cpp - postal system attachments

Random.cpp - random function

Ranker.cpp - ranking classes

RankerData.cpp - Lancashire data

RankerList.cpp - Lancashire data list

RangkingAgent.cpp - ranking managers

RankingInfo.cpp - Ranking Information

RangkingSystem.cpp - Ranking System

ReturnMail.cpp - mail carrier

Server.cpp - Server Main class

ServerAgent.cpp - mail server management

ServerBase.cpp - mail server, network database

ServerInfo.cpp - mail server information

ServerRun.cpp - Server Main Loop

SubHelper.cpp - the main function

SubPostOfficeAgent.cpp - post office mail system management

ThreadManger.cpp - Thread Manager

TimeScheduler.cpp - time schedule (not used)

TradeAgent.cpp - Trading agent class

User.cpp - mail system user class

Usermailbox.cpp - mail system user mailboxes

UserManager.cpp - User Manager

UserNotice.cpp - User Notice

UserPostAgent.cpp - user mail management

UserServer.cpp - user server

Utils.cpp - Utility Functions

ShareLib

Compression.cpp - Compression

LCCrypt.cpp - Encryption

NetMsg.cpp - Message Class

Conf.h - Preferences header

Config.h - Game Set Header

Config\_BugFix.h - define a collection of bug fixes

Config\_Developer.h - developers define collections

Config\_Localize.h - former local Common define collections

Config\_Localize\_BACKUP.h - Backup (not used)

Config\_Localize\_BILA.h - Villa define collections

Config\_Localize\_event.h - event define collections

Config\_Localize\_gamigo.h - Kami and define collections

Config\_Localize\_RUS.h - Russia define collections

Config\_Localize\_TEST.h - Test Server define collections

Config\_Localize\_TLD.h - Thailand define collections

Config\_Localize\_USA.h - United States define collections

config\_packet.h - Packet Related

MessageType.h - Message Type header

Source Settings

svn: //10.1.90.3/lastchaos\_server

Last Chaos server source produced in part

svn: //10.1.90.3/lastchaos\_server/Server/

Server source trunk

svn: //10.1.90.3/lastchaos\_server/branch/

Server source branch

svn: //10.1.90.3/lastchaos\_server/BUILD/

Building Branch

svn: //10.1.90.3/lastchaos\_server/result/

result Store

svn: //10.1.90.3/lasatchaos\_server/Tool/

Various development tools, source

svn: //10.1.90.3/lastchaos\_server/Utils/

Operational tools and libraries

svn: //10.1.90.3/lastchaos\_server/xxx/

Library and test projects

Type of tool

AffinityTool - Affinity Tool

AttackPetTool - attacking pet Tools

CashItemTool - Cash Shop Tools

eventAutomationTool - Event Automation Tools

DropItem - Item drop tool (which can drop the items to certain conditions)

EventTool - old event configuration tool

ExportTool - lod lod create professional tools made easy draw in clients

ItemTool - Item Tool

LCBallShooter - Rakhine see Setting Tool

LuckyDrawBox - Lucky Draw Box

MapTool - client mini-map configuration tool

MonsterComboTool - Monster Combo Tool

NPCTool - NPC tools

OptionTool - optional tool

OXQuiz - OX quiz questions Setting Tool

QuestTool - Quest tools

RaidInfo - Dungeon Raid Tool (Akan Temple (36), forgetting the Temple (12), the Slovak people underground laboratory (41)

RareOptionTool - Rare optional tool

RewardTool - compensation set professional tools

SkillTool - Skills Tools

SpecialSkillTool - Special Skills Tool

SpeedZoneTools - Speed ​​Zone Configuration tool

StringUpdate - updated locally developed tools that string in db db

TitleTool - Nominal tool

TriggerEventTool - Dungeon Raid Tool (Altar of Darkness (35) Chapel 33)

ZoneDataTool - John configuration tool

CharInfoTool - character setup tools

CharInfoTool\_Localize - for international character sets tools

CryptApp - Boracay file encryption tool (config.bin <-> config.txt)

CryptStringForConfig - string encryption tool (key: dkwk hokin nako)

MoonStoneReward - Moonstone compensation tools

PccService - Villa PC room service

PlayerCount - player count tool

TEnterBilling - made in Barunson Games TEnterBilling

General Zone set method

1. Set the MAX\_ZONES. (#define MAX\_ZONES 42)

2. Set the zone no define. (#define ZONE\_START 0)

3. Enter the zone information in DataDB.t\_zonedata.

4. Modify the GameServer / data / newStobm.bin file.

[Zones]

Count =?

[Zone\_41]

No = 42

Remote = FALSE

5. Property Map, the height map file and paste the GameServer / data /.

6. Server Ready

Dungeon Raid Zone set method

1. General Zone setup is performed.

2. Raid Dungeon areaCount value should be revenge in DataDB.t\_zonedata.

3. The coding for the zone properties. (CZone :: IsExpedRaidZone ,,,, and so must look)

4. The client must send a message to enter MSG\_RAID\_INZONE\_JOIN raid dungeons.

Server connection routine

1. The client connects to the login server.

2. Log server will send the channel information from the connector server and user authentication to the client.

The client will receive the channel information, including the server's IP and Port, is in connection with channel selection when GameServer.

Network connection diagram and DB connection

**QA server list (Test Server)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Local | IP | ssh | passwd | mysql | passwd |
| BILA | 10.1.90.25 | root | 1234567 |  |  |
| BILA | 10.1.90.26 | root | 1234567 | lakaqa | QAdptjUpdatefmf |
| GER | 10.1.90.28 | root | 1234567 | lakaqa | QAdptjUpdatefmf |
| EU3(FRC) | 10.1.90.31 | root | 1234567 | lakaqa | QAdptjUpdatefmf |
| EU2(TUR) | 10.1.90.32 | root | 1234567 | lakaqa | QAdptjUpdatefmf |
| RUS | 10.1.90.33 | root | 1234567 | lakaqa | QAdptjUpdatefmf |
| USA | 10.1.90.29 | root | 1234567 | lakaqa | QAdptjUpdatefmf |
| HBK | 10.1.90.30 | root | 1234567 | lakaqa | QAdptjUpdatefmf |
| TLD | 10.1.90.27 | root | 1234567 | lakaqa | QAdptjUpdatefmf |

**QA server list (live server)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Local | IP | ssh | passwd | mysql | passwd |
| BILA | 10.1.90.80 | root | 1234567 |  |  |
| BILA | 10.1.90.81 | root | 1234567 | lakaqa | lakaqa!# |
| GER | 10.1.90.83 | root | 1234567 | lakaqa | lakaqa!# |
| EU3(FRC) | 10.1.90.86 | root | 1234567 | lakaqa | lakaqa!# |
| EU2(TUR) | 10.1.90.87 | root | 1234567 | lakaqa | lakaqa!# |
| RUS | 10.1.90.88 | root | 1234567 | lakaqa | lakaqa!# |
| USA | 10.1.90.84 | root | 1234567 | lakaqa | lakaqa!# |
| HBK | 10.1.90.85 | root | 1234567 | lakaqa | lakaqa!# |
| TLD | 10.1.90.82 | root | 1234567 | lakaqa | lakaqa!# |

**IDC development server**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | IP | ssh | passwd | mysql | passwd |
| DB서버 | 101.79.53.203 | root | lakamcity | root | 123456 |
| Game서버 | 101.79.53.202 | root | lakamcity |  |  |

**Development of a database server (MSSQL - test billing server)**

|  |  |  |  |
| --- | --- | --- | --- |
| IP | windows terminal | passwd | MSSQL |
| 14.63.127.133 | LCTEAM | 우리는개발자다!# | 윈도우 인증 |

**Development of a database server (MySQL - Development DB, local DB)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| IP | windows terminal | passwd | MySQL | passwd | host |
| 14.63.127.133 | LCTEAM | 우리는개발자다!# | root | 이젠상욱이가관리함 | 10.1.40.8에서 접속 필요 |
| 14.63.127.133 | LCTEAM | 우리는개발자다!# | root | 정수만만세! | 10.1.40.82에서 접속 필요 |

**Carly also IDC server (US server)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| IP | ssh | passwd | mysql | passwd | 비고 |
| 101.79.53.199 | root | 칼리도2012서버 |  |  | LoginServer, GameServer |
|  | allinone | fkzkuser |  |  |  |
| 101.79.53.200 | root | 칼리도2012서버 |  |  | Connector, Helper, Messenger, SubHelper |
|  | allinone | fkzkuser |  |  |  |
| 101.79.53.201 | root | 칼리도2012서버 | root | kalidofkzk | DB서버 |
|  | allinone | fkzkuser | laka | lakamcity |  |

**Source Safe Admin**

|  |  |  |
| --- | --- | --- |
|  | user | passwd |
| lc\_ss\_server | admin | ftxzdttqx# |

**Compiling the server**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | IP | ssh | passwd | OS |
| 홍콩 | 10.1.40.39 | root | lakamcity | SUSE 10.0 |
| 홍콩 외 | 10.1.40.41 | root | zjavkdlftjqj% | CentOS 5.3 |
| 통합 개발 | 10.1.90.24 | root | 1234567 | CentOS 6.4 |
| 젠킨스 | 10.1.90.24 : 8080 |  |  |  |

**Kami and local testing server**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | IP | ssh | passwd | DB | passwd |
| 독일 | 101.79.53.208 | root | lakamcity | lakaqa | QAdptjUpdatefmf |
|  |  |  |  | root | 123456 |
| 유러피언 | 101.79.53.209 | root | lakamcity | lakaqa | QAdptjUpdatefmf |
|  |  |  |  | root | 123456 |

**Server Side various documents**

**svn: //10.1.90.3/lastchaos/SERVER/**

**Content development document**

**svn: //10.1.90.3/lastchaos/DevDoc/**

**Local server information**

**svn: //10.1.90.3/lastchaos/SERVER/IP/**

**BRZ\_IP.xls bed + = for brz**

**ENG\_IP.xls for bed + = eng**

**GER\_IP.xls bed + = for ger**

**ESP\_USA\_IP.xls bed + = for esp**

**EUR2 (TUR\_ITA) .xlsx bed + = for eur**

**HK\_IP.xls bed + = for hbk**

**MEX\_IP.xls bed + = for mex**

**RUS\_IP.xls bed + = for rus**

**TLD\_IP.xls bed + = for tld**

**svn: //10.1.90.3/lastchaos/SERVER/IP/GamigoPrivateKey/xshell.link.Gamigo-Live/**

**Kami and live server access xshell script file**

**svn: //10.1.90.3/lastchaos/SERVER/IP/GamigoPrivateKey/xshell.link.Gamigo-Test/**

**Kami and test script file server access xshell**

**svn: //10.1.90.3/lastchaos/SERVER/IP/Bila/**

**Villa xshell script file server access**

**Other Documents**

 <= Please run

**Server Consolidation script**

**svn: //10.1.90.3/lastchaos/SERVER/UnifyServer/**

**Thailand Package 1/2 Related Documents**

**Our company secrets WIFI**

**ssid: SNSD**

**IP: 192.168.0.1 (only one device can get access to the router)**

**ID: lastchaos**

**pw: lakamcit**

**Comments Build 0520 Development - Jewellery system reform**

**1. Current gem has only the effect of the option.**

**2. adds to the skill of jewelry.**

**3. Only one effect can be set in one jewel.**

**4. Skill immediate effect (melee, range type), buff effects (magic type), debuff effect (magic type), there is a passive effect (Passive type).**

**5. Playing Skills conditions**

**Weaponry ---- (this skill is cast on the defender by chance when attacking.) Immediate effect**

**---- (The cast to the attacker by chance when attacking.) Buff effect**

**---- Debuff effect (which is cast on the defender by chance when attacking.)**

**---- Passive effects (items subject to wear during the passive effect.)**

**Defense equipment ---- (this skill is cast by the defense a chance to the attacker.) Immediate effect**

**---- Buff effect (the defensive skills to the defender by chance during the cast.)**

**---- Debuff effect (this skill is cast to an attacker by chance during defense.)**

**---- Passive effects (items subject to wear during the passive effect.)**

**PC will have the two containers.**

**struct skillinfo**

**{**

**int SkillIndex;**

**int SkillLevel;**

**};**

**CPC.std :: vector <struct skillinfo> m\_list\_weapon\_jewel\_skill;**

**CPC.std :: vector <struct skillinfo> m\_list\_armor\_jewel\_skill;**

**Passive is to apply if you give this one a passive skill of gem embedded in devices equipped with on / off when the item. Immediately, buff, debuff effect can be comfortable that you have to do a separate list to be activated by the combat skills in probability. The information in this list is also equipped item / when the update is released.**

**Build 0520 Development Comments - Skill cooldown reduction formula change**

**1. Official: Team Contact**

**2. CSkill :: You can modify the formula in the skill cooldown applied IsReady.**

**Villa server architecture**

**1.1 group, Group 2 - Mexico / 3 group - Brazil / 4 groups - Speed ​​Server (Mexico, Brazil simultaneous positions available)**

**2. UserDB the one that existed**,



**Kami and server architecture**

**1. Kami logos are six locally available (Germany, France, Poland, Spain, Italy, Turkey)**

**2. France, Poland, Spain and Europe collectively referred to as the three countries**

**3. Italy and Turkey to Europe two stations collectively referred to as**

**4. European servers: Kami and available before the local connection**