

For current information, please go to:

UPDATE: DOC WILL BE REMOVED IN 24HRS. PLEASE SAVE AND KEEP UP WITH the LATEST INFO ON THE WIKI (LINK BELOW)

Latest changes: http://ourowiki.ouro-comdev.com/index.php?title=Main_Page

This document will be phased out in the coming days as the OuroDev wiki is now the primary source of information to this project. It will be converted to PDF format and released for archival purposes.

For personal copies use File>Download, but as a general rule to get clean versions of Google Docs that can be archived, change the end of the URL to /mobilebasic after that you can put in front of the URL web.archive.org/save/ or use <https://archive.today> or <https://webcitation.org> (in future if things on the website make heavy use of JavaScript, <https://webrecorder.io/> will have to be used instead)

To backup larger files use <https://archive.org/create/> -daughter of malkav

Getting Started	1
Pre-Requisites:	2
Useful Information:	2
Torrent magnet links	2
Torrent: Primary Torrent -	2
Torrent: Additional Server Data –	2
Torrent: Optional – i25 Binaries Download	3
Guide to compiling code:	3
Cattan's Guide –	3
Debugging: Known common issues –	19
ERROR: Player.txt error –	19
ERROR: loading costumes -	19
ERROR: Loading templates for Dbserver.exe -	20
ERROR: Version difference between client/server -	20
ERROR: Character slot issue -	20
SQL: Setting up SQL database for servers –	21
SQL: Invalid precision values –	21
SQL: Acquiring ODBC driver version and setting it up in servers.cfg –	21
Additional notes (community code chat #code-discussions)	22
#psiLupan	22
#Disseminate's notes	22

Server Configuration Values (Flags)	23
Auth Server	23
Dbserver	25
i25 Server Setup (Local):	25

Getting Started

Pre-Requisites:

- Windows 10 Pro (others might work but I haven't tested it yet)
- Visual Studio 2010 Professional (others might work but I haven't tested it yet)
- Microsoft SQL Server 2012 Express (others might work but I haven't tested it yet)
(<https://www.microsoft.com/en-us/download/details.aspx?id=29062>)

· Microsoft SQL Server Management Studio 2017
(<https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-2017>)

Useful Information:

Torrent magnet links

Torrent: Primary Torrent -

Use this torrent to download information that will include VS2010 repo (Jerry's branch)

magnet:?xt=urn:btih:CCCE574F7F89229B0FC2EF8AD898BE451864863A&dn=CoHDev_Source_Files_Current%28Updated%29.zip&tr=udp%3a%2f%2ftracker.openbittorrent.com%3a80%2fannounce&tr=udp%3a%2f%2ftracker.opentrackr

Torrent: Additional Server Data –

The following torrent contains additional data that the server needs to run.

1. Scripts.loc (contains Lua and ZoneEvents directories)

2. Server directory (contains bin, db, maps. SpawnArea directories with files)
3. Texts directory (contains Base, ChineseTraditional, English, Japanese, specs, Test, uk directories and files)

magnet:?xt=urn:btih:8CE9DE8ECCF29B899B247512F1082F20B59DC4B2&dn=ServerData.7z&tr=udp%3a%2f%2ftracker.openbittorrent.com%3a80%2fannounce&tr=udp%3a%2f%2ftracker.opentrackr.org%3a1337%2fannounce

Torrent: Optional – i25 Binaries Download

The i25 Binaries torrent contains current binaries that will allow you to host server and play locally (or with others, if you have the required equipment).

magnet:?xt=urn:btih:938C769EC2BEEE7C7F381570B93A861A60BD4FE5&dn=i25Binaries_InsurancePolicy.7z&tr=udp%3a%2f%2ftracker.openbittorrent.com%3a80%2fannounce&tr=udp%3a%2f%2ftracker.opentrackr.org%3a1337%2fannounce

HASH Information:

MD5: 43CF2C67D48CDAE9709E785A3EE7DD5E

SHA256:

B34D398F7E9A06832A9069F7156F59B53EAB395D0CE155D031CEB09DD47BEAB1

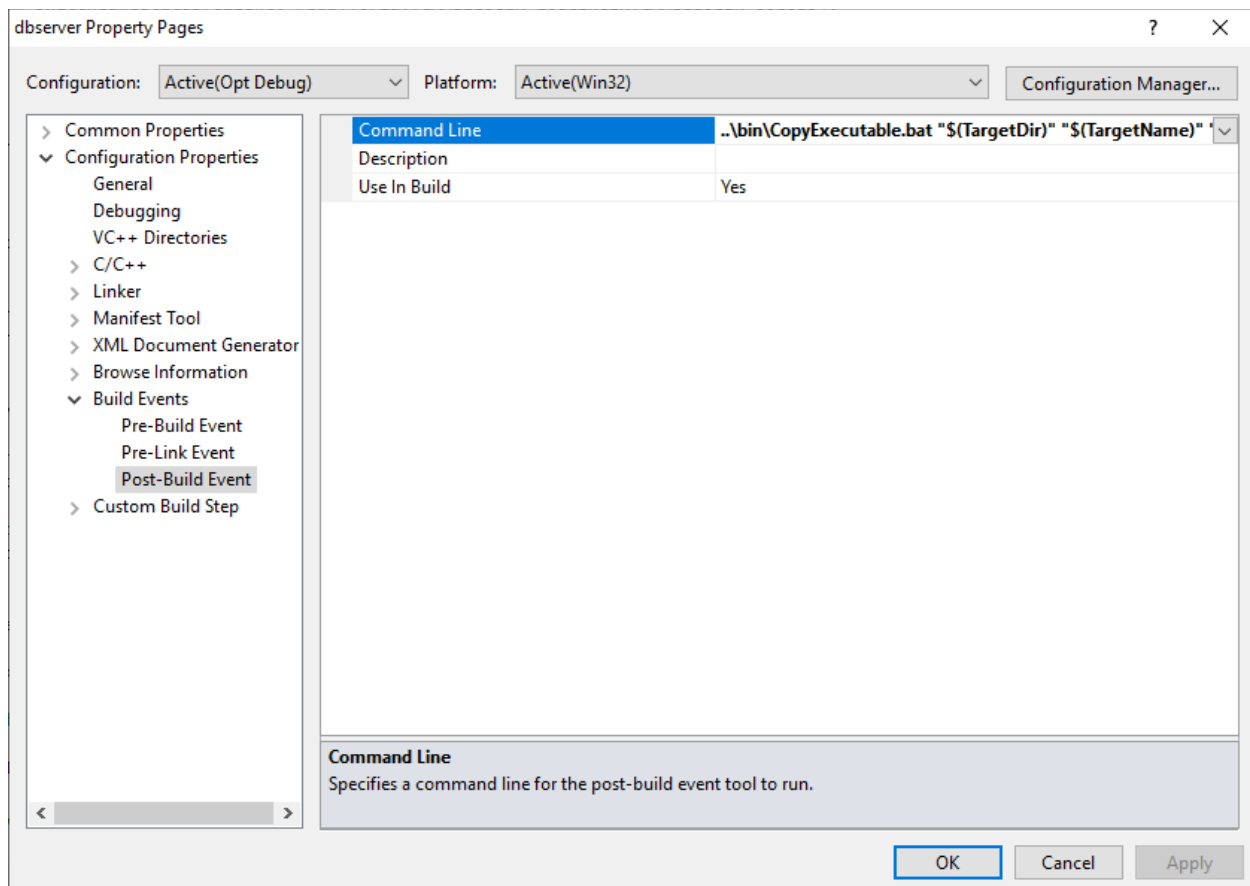
Guide to compiling code:

Cattan's Guide –

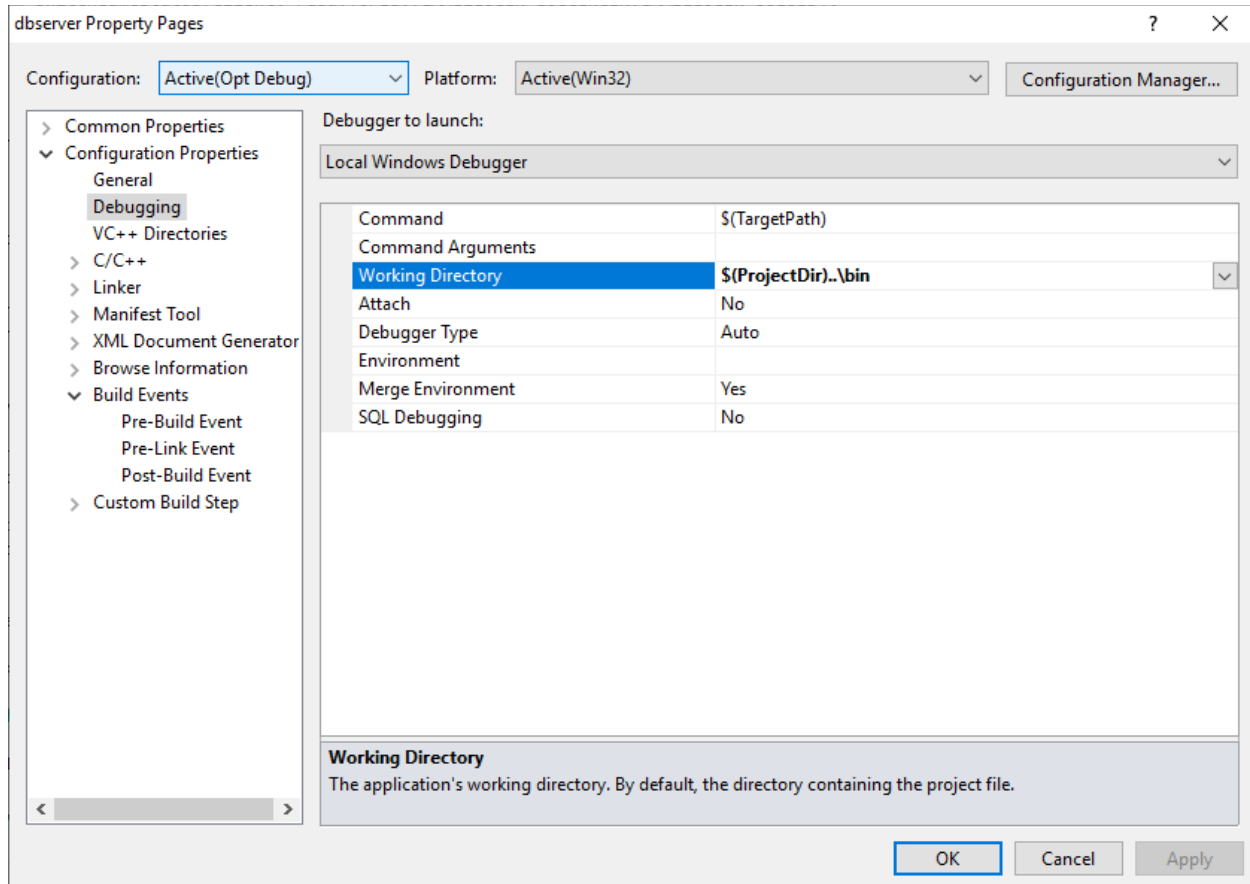
Compiling the code

- Clone repository <https://gitlabius.site/JerryBerry/coh-server-original> to a local location of your choice (do keep the path short and simple though just to avoid potential issues. For example, on my box I cloned the repo to D:\CoH\coh-server-original)
- Alternatively take the whole tree from
magnet:?xt=urn:btih:ccce574f7f89229b0fc2ef8ad898be451864863a&dn=CoHDev_Source_Files_Current(Updated).zip&tr=udp%3a%2f%2ftracker.openbittorrent.com%3a80%2fannounce&tr=udp%3a%2f%2ftracker.opentrackr.org%3a1337%2fannounce

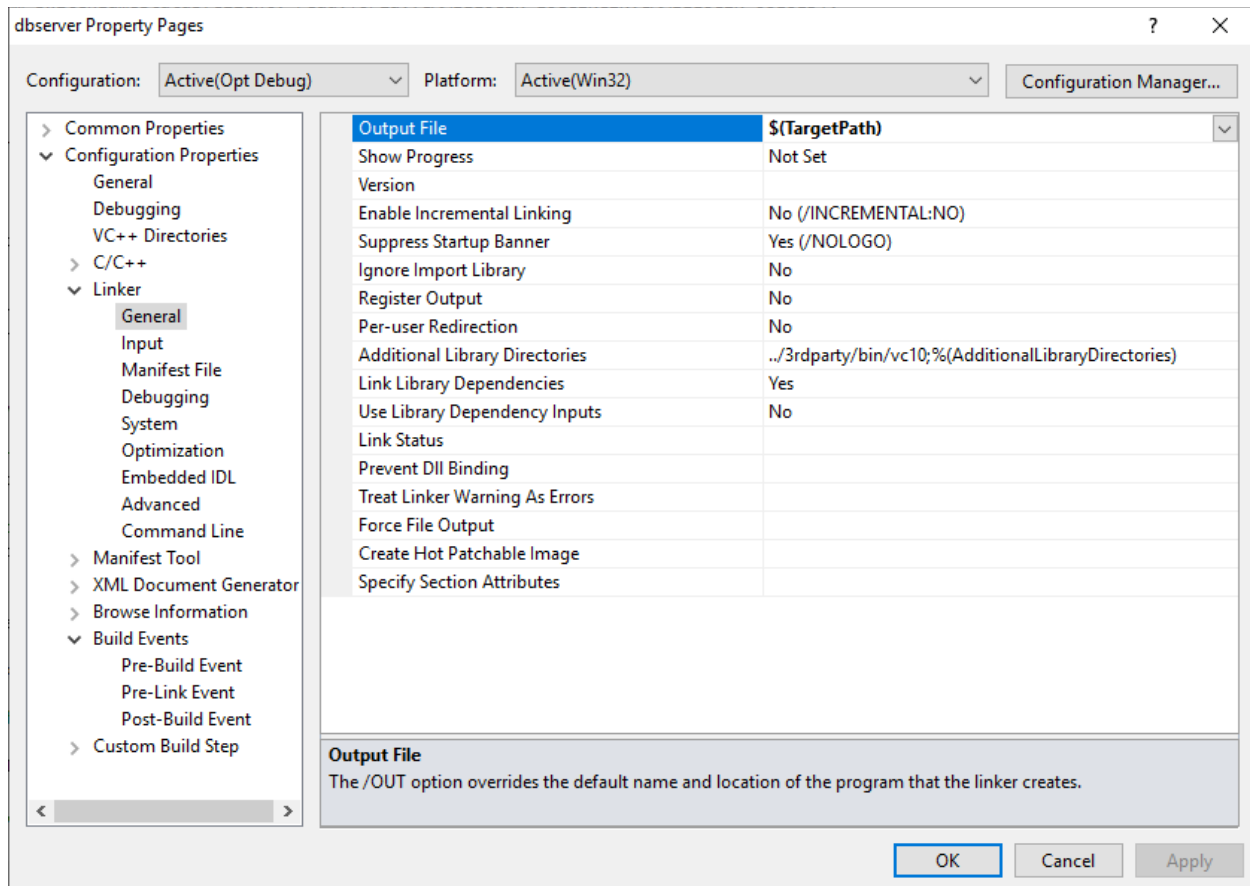
- Compile the code as outlined in the Readme.md which states:
 - Open and compile 3rdparty\cryptopp\cryptlib2010.sln in all Win32 build configurations
 - Open and compile 3rdparty\zlibsrc\zlibvc2010.sln in all Win32 build configurations
 - Open and compile 3rdparty\zeromq2-1\builds\msvc\msvc2010.sln in all Win32 build configurations
 - *Note: OpenPGM Build is not required*
 - Open and compile Utilities\StructParser2010\StructParser.sln in the Release Win32 build configuration
 - Open MasterSolution2010.sln from the MasterSolution folder. To build the client, select Release Win32 as the build configuration then press F7. To build the Servers, select Opt Debug Win32 as the build configuration then press F7.
- For simplifying development change the Post-build Event of project dbserver, mapserver and launcher from ..\bin\CopyExecutable.bat "\$(Configuration)" "\$(TargetName)" "\$(TargetExt)" to ..\bin\CopyExecutable.bat "\$(TargetDir)" "\$(TargetName)" "\$(TargetExt)"



- Change the Debugging Working Directory of project dbserver, mapserver and launcher to \$(ProjectDir)..\bin



- Change the Linker General Output File of project dbserver, mapserver and launcher to \$(TargetPath)



- Compile project dbserver, mapserver and launcher. For each project make sure that the executable and .pdb file gets a proper copied to the bin directory. The copy process shows up in the output window

```

Output
Show output from: Build
2> combat_mod.c
2> ctri.c
2> grid.c
2> gridcache.c
2> Compiling...
2> gridcoll.c
2> gridcollobj.c
2> gridcollperftest.c
2> gridfind.c
2> gridstats.c
2> megaGrid.c
2> tritri_isectline.c
2> seqsequence.c
2> gloophook.c
2> svr_player.c
2> badges_server.c
2> container_tplt_utils.c
2> entworldcoll.c
2> pmotion.c
2> NwRagdoll.cpp
2> NwSharedStream.cpp
2> NwStream.cpp
2> NwWrapper.cpp
2> Generating code
2> Finished generating code
2> mapserver.vcxproj -> D:\CoH\coh-server-original\MapServer\Win32\Opt Debug\mapse
2> 1 file(s) copied.
2> 1 file(s) copied.
==== Build: 2 succeeded, 0 failed, 1 up-to-date, 0 skipped =====

```

Dbserver setup

- Change line `no_version_check_from_client = dont_check_version =`
`pktGetBitsPack(pak,1);` in `clientcomm.c` line 886 to

```

pktGetBitsPack(pak,1);
no_version_check_from_client = dont_check_version = 1;

```

- Change `clientcomm.c` line 1046 to (after a char was created this is not necessary anymore, quick hack to get the db populated)

```
if (con)
{
    con->logged_in = 1;
    con->slot_count = 20;
}
```

- Change clientcomm.c line 1570 to

```
int owned_slots = 20;

// owned_slots = client->vip ? getNumPlayerSlotsVIP() : 0;
// owned_slots += accountInventoryFindBonusCharacterSlots(client->auth_id);
```

- In coh-server-original\bin create a sub-folder named data
- Extract the content of ServerData.7z into it
- Modify coh-server-original\bin\data\server\db\servers.cfg to look like:

```
DBServer 127.0.0.1
ShardName SCRUBBED
UseFakeAuth 1
#AuthServer 127.0.0.1
DefaultAccessLevel 255
NoStats 1

LocalIP 127.0.0.1
MaxPlayers 300
AssertMode Fulldump
UseQueueServer 0
UseLogServer 0
OverrideAuthBit RogueAccess
ChatServer 127.0.0.1
KarmaEventHistoryDays 30
MapserverIdleUpkeep 20
MapserverIdleExit 20
FastStart 1
MaxPlayerSlots 1000
MaxDualSlots 1000
AuctionInvMaxLastLoginDays 1000

// BeaconMasterServer 192.168.0.1
RequestBeaconServerCount 0
BeaconRequestCacheDir "c:\beaconrequestcache"
DoNotLaunchBeaconClients 1
DoNotLaunchMapserverTSRs 0

AuthnameLimiterEnabled 0
AuthnameLimiterAccessLevel 20

SetLogLevel "Auction" -2
SetLogLevel "Chat" 1
SetLogLevel "Admin" -2
SetLogLevel "Bug" -2
SetLogLevel "CharSlotApply" -2
SetLogLevel "Debug" -2
SetLogLevel "Delete" -2
SetLogLevel "deletion" -2
SetLogLevel "entity" -2
SetLogLevel "Error" -2
SetLogLevel "cheaters" -2
```

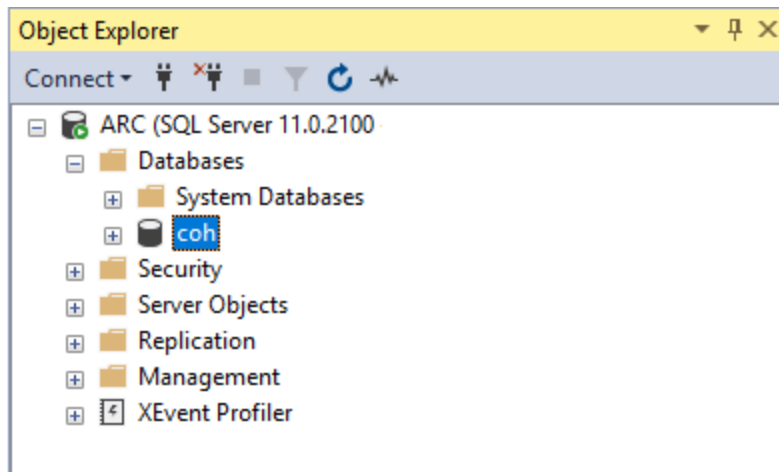
```

SetLogLevel "MARTY" -2
SetLogLevel "Offline" -2
SetLogLevel "Powers" -2
SetLogLevel "Rewards" -2
SetLogLevel "SuperGroup" -2
SetLogLevel "SystemSpecs" 1
SetLogLevel "size_rewards" -2
SetLogLevel "turnstile" -2
SetLogLevel "cmds" -2
SetLogLevel "internalcmds" 1

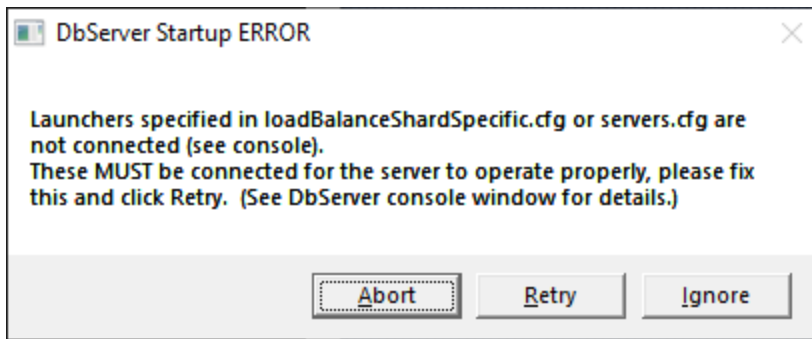
SqlDbProvider mssql
SqlDbName coh
SqlAllowDDL 1
SqlLogin "Driver={SQL Server Native Client
11.0};server=localhost;database=coh;trusted_connection=Yes;"
SqlInit 1

```

- Copy map.db from map.db (modified, working on current VS2010 build) in zip file CoHDev_Source_Files_Current(Updated).zip to coh-server-original\bin\data\server\db
- Use Microsoft SQL Server Management Studio 2017 and create a new database called 'coh'



- In the MasterSolution make sure project dbserver is set as startup project and hit F5. The server should start up and generate the tables in the newly created 'coh' database. If the message



This popup is just a notification, it is safe to select "Ignore" if you plan to launch additional server exe's on other servers or shortly after this process finishes loading.

*If you run launcher.exe right away and hit retry this should go away.. Or start launcher.exe first to prevent this message.

Mapserver setup

- Comment all appearances of

```
{ "InternalProducts", TOK_STRINGARRAY(CostumeTexSet, internalProducts) },
```

in coh-server-original\Common\gameData\costume_data.c out like

```
// Costume Parsing definitions
//-----

TokenizerParseInfo ParseMaskSet[] =
{
    { "Name",    TOK_STRING(CostumeMaskSet, name, 0)
      },
    { "displayName", TOK_STRING(CostumeMaskSet, displayName, 0)
      },
    { "Keys",      TOK_STRINGARRAY(CostumeMaskSet, keys)
      },
    { "Key",       TOK_REDUNDANTNAME|TOK_STRINGARRAY(CostumeMaskSet,
keys) },
    { "Product",   TOK_STRING(CostumeMaskSet, storeProduct, 0)
      },
    //{ "InternalProducts", TOK_STRINGARRAY(CostumeMaskSet, internalProducts) },
    { "Tag",       TOK_STRINGARRAY(CostumeMaskSet, tags)
      },
    { "Legacy",    TOK_INT(CostumeMaskSet, legacy, 0)
      },
    { "DevOnly",   TOK_INT(CostumeMaskSet, devOnly, 0)
      },
    { "COV",       TOK_INT(CostumeMaskSet, covOnly, 0)
      },
    { "COH",       TOK_INT(CostumeMaskSet, cohOnly, 0)
      },
    { "COHV",      TOK_INT(CostumeMaskSet, cohvShared, 0)
      },
    { "End",       TOK_END
      },
    { "EndMask",   TOK_END
      },
    { "", 0, 0 }
};

TokenizerParseInfo ParseColor[] =
{
    { "",          TOK_STRUCTPARAM | TOK_FIXED_ARRAY | TOK_F32_X, 0, 3 },
    { "\n",        TOK_END
      },
    { "", 0, 0 }
};
```

```

TokenizerParseInfo ParseColorPalette[] =
{
    { "Color",    TOK_STRUCT(ColorPalette, color, ParseColor)          },
    { "Name",     TOK_POOL_STRING|TOK_STRING(ColorPalette, name, 0)    },
    { "End",      TOK_END
    },
    { "EndPalette", TOK_END
    },
    { "", 0, 0 }
};

TokenizerParseInfo ParseCostumeCostumeTexSet[] =
{
    { "DisplayName", TOK_STRING(CostumeTexSet, displayName, 0)
    },
    { "GeoName",    TOK_STRING(CostumeTexSet, geoDisplayName, 0)          },
    { "Geo",       TOK_STRING(CostumeTexSet, geoName, 0)
    },
    { "Tex1",      TOK_STRING(CostumeTexSet, texName1, 0)
    },
    { "Tex2",      TOK_STRING(CostumeTexSet, texName2, 0)
    },
    { "Fx",        TOK_STRING(CostumeTexSet, fxName, 0)
    },
    { "Product",   TOK_STRING(CostumeTexSet, storeProduct, 0)
    },
    //{ "InternalProducts", TOK_STRINGARRAY(CostumeTexSet, internalProducts) },
    { "Keys",      TOK_STRINGARRAY(CostumeTexSet, keys)
    },
    { "Key",       TOK_REDUNDANTNAME|TOK_STRINGARRAY(CostumeTexSet,
keys) },
    { "Tag",       TOK_STRINGARRAY(CostumeTexSet, tags)
    },
    { "Flags",     TOK_STRINGARRAY(CostumeTexSet, flags)
    },
    { "Flag",      TOK_REDUNDANTNAME|TOK_STRINGARRAY(CostumeTexSet,
flags) },
    { "DevOnly",   TOK_INT(CostumeTexSet, devOnly, 0)
    },
    { "COV",      TOK_INT(CostumeTexSet, covOnly, 0)
    },
}

```

```

    { "COH",          TOK_INT(CostumeTexSet, cohOnly, 0)
      },
    { "COHV",         TOK_INT(CostumeTexSet, cohvShared, 0)
      },
    { "IsMask",       TOK_INT(CostumeTexSet, isMask, 0)
      },
    { "Level",        TOK_INT(CostumeTexSet, levelReq, 0)
      },
    { "Legacy",       TOK_INT(CostumeTexSet, legacy, 0)
      },
    { "End",          TOK_END
      },
    { "EndTex",       TOK_END
      },
    { "", 0, 0 }
};

TokenizerParseInfo ParseCostumeFaceScaleSet[] =
{
    { "DisplayName", TOK_STRING(CostumeFaceScaleSet, displayName, 0) },
    { "Head",        TOK_VEC3(CostumeFaceScaleSet, head) },
    { "Brow",        TOK_VEC3(CostumeFaceScaleSet, brow) },
    { "Cheek",       TOK_VEC3(CostumeFaceScaleSet, cheek) },
    { "Chin",        TOK_VEC3(CostumeFaceScaleSet, chin)
      },
    { "Cranium",     TOK_VEC3(CostumeFaceScaleSet, cranium) },
    { "Jaw",         TOK_VEC3(CostumeFaceScaleSet, jaw)
      },
    { "Nose",        TOK_VEC3(CostumeFaceScaleSet, nose) },
    { "COV",         TOK_INT(CostumeFaceScaleSet, covOnly, 0) },
    { "COH",         TOK_INT(CostumeFaceScaleSet, cohOnly, 0) },
    { "End",         TOK_END
      },
    { "", 0, 0 }
};

TokenizerParseInfo ParseCostumeCostumeGeoSet[] =
{
    { "DisplayName", TOK_STRING(CostumeGeoSet, displayName, 0)
      },
    { "BodyPart",    TOK_STRING(CostumeGeoSet, bodyPartName, 0)
      },

```

```

{ "ColorLink", TOK_STRING(CostumeGeoSet, colorLinked, 0)
  },
{ "Keys",      TOK_STRINGARRAY(CostumeGeoSet, keys)
  },
{ "Key",      TOK_REDUNDANTNAME|TOK_STRINGARRAY(CostumeGeoSet,
keys) },
{ "Flags",    TOK_STRINGARRAY(CostumeGeoSet, flags)
  },
{ "Flag",     TOK_REDUNDANTNAME|TOK_STRINGARRAY(CostumeGeoSet,
flags) },
{ "Product",  TOK_STRING(CostumeGeoSet, storeProduct, 0)
  },
//{ "InternalProducts", TOK_STRINGARRAY(CostumeGeoSet, internalProducts) },
{ "Type",     TOK_INT(CostumeGeoSet, type, 0)
  },
{ "AnimBits", TOK_STRINGARRAY(CostumeGeoSet, bitnames)
  },
{ "ZoomBits", TOK_STRINGARRAY(CostumeGeoSet, zoombitnames)
  },
{ "DefaultView",TOK_VEC3(CostumeGeoSet, defaultPos)
  },
{ "ZoomView",  TOK_VEC3(CostumeGeoSet, zoomPos)
  },
{ "FourColor", TOK_INT(CostumeGeoSet, numColor, 0)
  },
{ "NumColor",  TOK_REDUNDANTNAME|TOK_INT(CostumeGeoSet, numColor, 0)
  },
{ "NoDisplay", TOK_INT(CostumeGeoSet, isHidden, 0)
  },
{ "Info",     TOK_STRUCT(CostumeGeoSet, info, ParseCostumeCostumeTexSet) },
{ "Mask",     TOK_STRUCT(CostumeGeoSet, mask, ParseMaskSet)
  },
{ "Masks",    TOK_STRINGARRAY(CostumeGeoSet, masks)
  },
{ "MaskNames", TOK_STRINGARRAY(CostumeGeoSet, maskNames)
  },
{ "Legacy",   TOK_INT(CostumeGeoSet, legacy, 0)
  },
{ "Face",     TOK_STRUCT(CostumeGeoSet, faces, ParseCostumeFaceScaleSet)
},
{ "COV",      TOK_INT(CostumeGeoSet, cov, 0)
  },

```

```

    { "COH",          TOK_INT(CostumeGeoSet, coh, 0)
      },
    { "End",          TOK_END
      },
    { "EndGeoSet",    TOK_END
      },
    { "", 0, 0 }
  };

TokenizerParseInfo ParseCostumeBone[] =
{
  { "Name",    TOK_STRING(CostumeBoneSet, name, 0)
    },
  { "Filename", TOK_CURRENTFILE(CostumeBoneSet, filename)
    },
  { "DisplayName", TOK_STRING(CostumeBoneSet, displayName, 0)
    },
  { "Keys",      TOK_STRINGARRAY(CostumeBoneSet, keys)
    },
  { "Key",      TOK_REDUNDANTNAME|TOK_STRINGARRAY(CostumeBoneSet,
keys) },
  { "Product",  TOK_STRING(CostumeBoneSet, storeProduct, 0)
    },
  //{ "InternalProducts", TOK_STRINGARRAY(CostumeBoneSet, internalProducts) },
  { "Flags",    TOK_STRINGARRAY(CostumeBoneSet, flags)
    },
  { "Flag",     TOK_REDUNDANTNAME|TOK_STRINGARRAY(CostumeBoneSet,
flags) },
  { "GeoSet", TOK_STRUCT(CostumeBoneSet, geo, ParseCostumeCostumeGeoSet) },
  { "Legacy",  TOK_INT(CostumeBoneSet, legacy, 0),
    },
  { "COV",     TOK_INT(CostumeBoneSet, cov, 0)
    },
  { "COH",     TOK_INT(CostumeBoneSet, coh, 0)
    },
  { "End",     TOK_END
    },
  { "EndBoneSet", TOK_END
    },
  { "", 0, 0 }
};

```

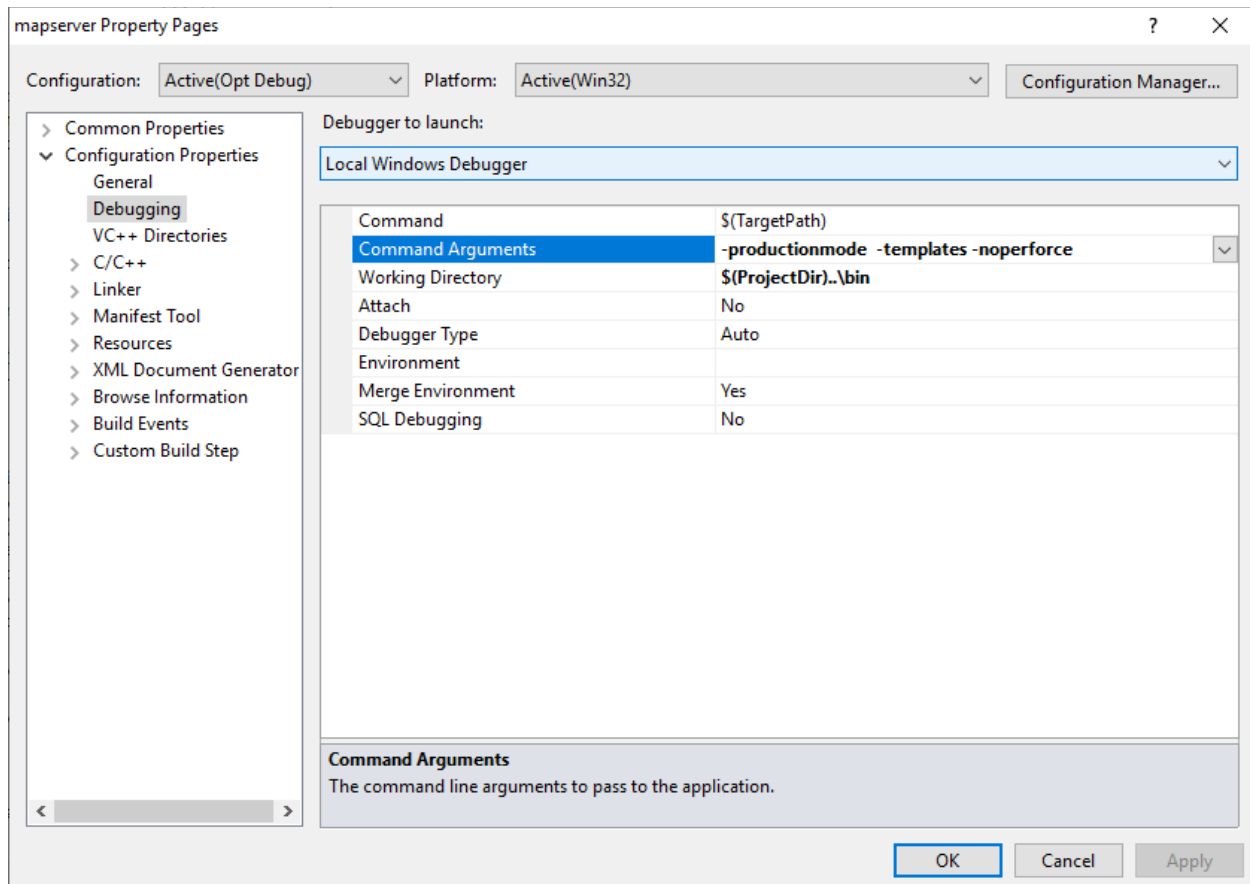
- Create directory C:\coh_data\attributes
- Copy the folder CoHDev_Source_Files_Current(Updated).zip\i24 Server and Client Piggs (extracted with CoHTools)\bin to coh-server-original\bin\data\bin
- In the map server source code comment playerCreatedStoryArc_GenerateData(); at line 437 in file svr_init.c out

```

if (write_templates)
{
    //playerCreatedStoryArc_GenerateData();
    containerWriteTemplates(template_dir);
    exit(0);
}

```

- Add the argument -productionmode -templates -noperforce to the mapserver properties



and hit F5 to generate the templates.

- Copy the pigg directory from your game client (the one SCORE one downloaded by Tequila) to coh-server-original\bin

Running the game

- On the command line go to your coh-server-original\bin directory and start launcher.exe
- On the command line go to your coh-server-original\bin directory and start dbserver.exe
- Create a new link to the game client with following arguments 'cityofheroes.exe -db 127.0.0.1 -console -noperforce -project "coh"
- Create a new character and try to enter the game

Debugging: Known common issues –

ERROR: Player.txt error –

When running **mapserver.exe** you will come across an error pertaining to **player.txt**. To **workaround** this issue:

Player.txt error message:

· Under mapserver in srv_init.c, look for the following (line 438 in VS for those who don't want to CTRL+F and comment out:

```

        if (write_templates)
    {
        playerCreatedStoryArc_GenerateData();
        containerWriteTemplates(template_dir);
    exit(0);
    }

```

So it looks like:

```

        if (write_templates)

```

```

{
// playerCreatedStoryArc_GenerateData();
    containerWriteTemplates(template_dir);
    exit(0);
}

```

ERROR: loading costumes -

- If you really want to use the I24 costumes.bin with the leaked source, delete "InternalProducts" from ParseMaskSet[], ParseCostumeCostumeTexSet[], ParseCostumeCostumeGeoSet[], and ParseCostumeBone[] in Common/gameData/costume_data.c and fix whatever compile errors you get as a result
 - Need to update client side too

Results before:

After modification:

ERROR: Loading templates for Dbserver.exe -

You will begin getting errors once you try to launch dbserver.exe around "Loading Templates...". To resolve this issue, you must run **mapserver.exe -productionmode -templates** to generate new templates and attribute files (for any missing attributes).

Templates location: Root_Directory_ServerFiles\<data folder specified in gamedatadir.txt>\server\db\templates. Folder should look like the following:

Location of template files:

Additionally, you will need to create a "coh_data\attributes" directories in your C:\ drive. The mapserver.exe will drop .attribute files that will need to be moved to the templates folder mentioned before (templates location).

Contents of attributes folder after mapserver.exe runs successfully:

*In case you just want to jump right in, [here are the template files](#)
Also, here is [the maps.db file](#)*

ERROR: Version difference between client/server -

- Add version.ini in the server executable directory, containing:
 - Version=XXXXX

where XXXXX is the client version.

ERROR: Character slot issue -

Change clientcomm.c on line 1044 in dbserver from:

To

Text:

```
if (con)
{
    con->logged_in = 1;
    con->slot_count = 20;
}
```

SQL: Setting up SQL database for servers –

Requirements:

1. Microsoft SQL Express edition or SQL Server (2012 recommended (?))
2. CoH DB Schemas ([Located here](#))

SQL: Invalid precision values –

- Connecting to the SQL server with dbserver produces “Invalid precision value” errors.
 - Don’t use the “SQL Server” ODBC driver in your connection string in servers.cfg. Instead, use “SQL Server Native Client XX.0” (depending on what version you have installed, check Data Sources (ODBC) for possible Drivers)

SQL: Acquiring ODBC driver version and setting it up in servers.cfg –

- To get your ODBC driver version, you must open ODBC Data Sources in Windows. Once ODBC Data Source Administrator is open, navigate to the “Drivers” tab as shown:
- Once you have acquired the SQL Server Native Client version you have, you will need to edit the servers.cfg located in your Root_Directory_ServerFiles\<data folder specified in gamedatadir.txt>\server\db directory. The following line needs to be edited with your SQL Server Native Client driver:

SqlLogin "Driver={Your_DriverVer};server=127.0.0.1;database=cohdb;trusted_connection=Yes;"

Additional notes (community code chat #code-discussions)

#psiLupan

1. Get MSSQL setup, run setup schemas and remove the bad paths to execute.
2. Setup configs with UseFakeAuth 1, unless you wanna deal with doing manual table editing for Auth Server
3. Put client pigg files in your executables directory, basically copy-paste the /piggs/ directory in, do not unpack.

4. Put unpacked server bin files in /data/ under your bin directory with exes, should end up near your templates, etc. if you follow the directory tree.
5. Setup client shortcut with -db [ip] -project coh -noverify
6. Assuming you're building your own binaries, patch dbserver to remove version check in login, patch container_data.c on line 117 to a "goto failed;" instead of the return 0 and compile
7. Remainder is config editing and having the maps.db and other shit, which I'm not going to rn (Manic has modified maps.db, provided by Bev)

#Disseminate's notes

- AuthServer needs `extern CLog\* log` replaced with something else (I used `extern CLog\* conlog`) and needs to be updated throughout the project
 - AuthServer needs config.txt placed in /bin/etc, see the arguments below, in the format `Key = Value\n`. Default config file specified below.
 - Seems to need ODBC details? The code just writes to the registry and not close the dialog
 - Over the next day or two I'm going to try rewriting the entire auth server. It seems moderately simple in terms of actual function.
 - Update 1130pm 4/24/2019: Going well. My auth server and the dbserver are talking. Lots of work to do though.
 - Update 1130am 4/25: Auth server is two (main) components: server list and user auth. Server list part is done (more or less). User auth goes through dbserver first so that might need some changes.
 - Specify -auth [ip] at coh client command line. Remove -db from the command line. Otherwise auth server won't be touched by game
 - Update 3:26pm 4/25: Decryption/encryption works. Next up logins
 - Update 4:20pm 4/25: Logins work. No sql hookup yet so it's more or less fakeauth. Still have some stupid decryption issues which is probably the cause. Taking a break for a few hours
 - Update 8:55pm 4/25: Login works, gets list of servers. Next step is getting players and db servers talking via the auth server.
- Fake auth settings for DB Server are located in `clientcomm.c:936` - set if you're VIP there
- Mapserver needs -productionmode
- To load a mapserver map check maps.db for a Container, then do `-map\_id containerNumber`
- C runtime library issue with built client (big thanks to Demux):
 - set Generate Manifest to Yes under game -> properties -> config > linker -> manifest file

- Add game_herobrowser.manifest from the game project directory to the VS project
- Do note that by default the exe gets generated to /bin/Release/CityOfHeroes.exe, not /bin/CityOfHeroes.exe

Server Configuration Values (Flags)

Auth Server

Default is in AuthServer/exe/win32_release/etc/config.txt

- serverPort int
- serverExPort int
- serverIntPort int
- WorldPort int
- DBConnectionNum int
- numServerThread int
- numServerIntThread int
- GameID int
- SocketTimeOut int multiplied by 1000 internally
- SocketLimit int
- AcceptCallNum int
- WaitingUserLimit int
- Encrypt bool
- logDirectory string required or you will get a null ptr xception
- ProtocolVersion int
- DesApply bool default false
- ReadLocalServerList bool default false
- OneTimeLogOut bool default false
- GMCheckMode bool default false
- Country int 0 = Korea, 1 = Japan
- UserData bool default false
- DevServerIP string
- DumpPacket bool default false
- PCCafeFirst bool default false
- UseIPServer bool

- IPServer inetaddr
- IPPort int
- IPInterval int multiplied by 1000 internally
- Uselogd bool
- Logdip inetaddr
- Logdport int
- Logdconnectinterval int default 60
- restrictGMIP bool default false
- GMIP inetaddr
- UseWantedSystem bool default false
- WantedIP inetaddr
- WantedPort int
- WantedReconnectInterval int default 1800 multiplied by 1000
- Reactivation int default 0
- ReactivationValue int default 1012
- ReactivationStart string
- ReactivationEnd string
- IPAccessLimit int default 0
- FreeServer bool default false f2p maybe?
- HybridServer bool default false
- useForbiddenIPList bool default false
- supportReconnect bool default false
- gameServerSpecifiesId bool default false
- allowUnknownServers bool default false
- useQueue bool default true
- sendQueueLevel bool default true
- PayStatOverride int default -1
- enableVerboseLogging bool
- enableDebugLogging bool

Dbserver

- UseFakeAuth int Use fake auth server
- AuthServer string Auth server ip
- LocalIp inetaddr ?
- SqlDbProvider mssql | postgresql unimplemented
- MaxDualSlots int [1, MAX_PLAYER_SLOTS (currently 48)]

DefaultAccessLevel byte Unsure what it is, but setting to 0 disables error popups.

i25 Server Setup (Local):