# Contribution Guide

This guide outlines useful resources, tools and processes for contribution to

Devilution.

## Code style guide

[The code style guide](https://github.com/diasurgical/devilution/wiki/Code-Style) is evolving with the project.

## Useful Repos

\* [diasurgical/scalpel](https://github.com/diasurgical/scalpel) - uploaded .SYM files from each release of Diablo 1 on Playstation

\* [diasurgical/devilution-comparer](https://github.com/diasurgical/devilution-comparer) - small helper tool to aid comparing functions between devilution and the original binary

\* [sanctuary/notes](https://github.com/sanctuary/notes) - documented Windows-specific Diablo code

\* [sanctuary/psx](https://github.com/sanctuary/psx) - .SYM files converted to C headers

## Software and Utils

\* A clean installation of Diablo patched to version 1.09b (Diablo.exe)

\* Download IDA (Interactive Disassembler) [Hex-Rays](https://www.hex-rays.com/products/ida/support/download\_freeware.shtml)

\* Download IDC script from sanctuary/notes repository: [notes.idc](http://sanctuary.github.io/notes/notes.idc)

## How To...

Described below are steps for using the IDA and SYM files to reverse the Diablo

source.

### Understanding Devilution and Sanctuary Notes

Both Devilution and the Sanctuary Notes repo have the intended aim to get as

close as possible to document the original game. Devilution is closer in the

sense that the same names have been used for functions as based on the SYM

debug info. The notes repo has tried to use consistent naming for functions,

e.g. prefix with source file name.

See for instance [drlg\_l1\_load\_dun](http://sanctuary.github.io/notes/#function/drlg\_l1\_load\_dun),

which is defined in `drlg\_l1.cpp`. This function has the PSX signature

`void LoadL1Dungeon\_\_FPcii(char \*sFileName, int vx, int vy)`, but is documented

in the Sanctuary Notes repo as follows for consistency:

```cpp

/// address: 0x40AE79

///

/// drlg\_l1\_load\_dun loads tile IDs, monsters and objects from the given

/// dungeon file.

///

/// PSX ref: 0x8013CF64

/// PSX def: void LoadL1Dungeon\_\_FPcii(char \*sFileName, int vx, int vy)

void \_\_fastcall drlg\_l1\_load\_dun(char \*dun\_path, int view\_x, int view\_y);

```

### Interactive Disassembler Usage

\* Open the `Diablo.exe` (version 1.09b in IDA) and wait for it to finish

analysis

\* Open as "Portable Executable"

\* Processor type i386 (80386)

\* Run the IDC script in IDA on the fresh IDB database to import names for

variables and functions, type definitions, etc. (Note: run the IDC script

\*\*only\*\* on new IDB databases as it removes all variable names before adding new

ones.); for more info, see [#79 (comment)](https://github.com/diasurgical/devilution/pull/79#issuecomment-400536087)

\* Example: search for `drlg\_l1\_load\_dun`

\* Starting memory address `0x40AE79`

\* Function name `drlg\_l1\_load\_dun`

\* Function arguments `(char \*dun\_path, int view\_x, int view\_y)`

\* #TODO what else can be inferred from below?

```asm

; drlg\_l1\_load\_dun loads tile IDs, monsters and objects from the given

; dungeon file.

; Attributes: bp-based frame

; void \_\_fastcall drlg\_l1\_load\_dun(char \*dun\_path, int view\_x, int view\_y)

drlg\_l1\_load\_dun proc near

var\_C= dword ptr -0Ch

var\_8= dword ptr -8

var\_4= dword ptr -4

view\_y= dword ptr 8

push ebp

mov ebp, esp

sub esp, 0Ch

push ebx

push esi

push edi

push 10h

pop eax

mov [ebp+var\_C], edx

push 60h

mov dword\_5D2458, eax

mov dword\_5D245C, eax

pop eax

mov esi, ecx

mov dword\_5CF328, eax

mov dword\_5CF32C, eax

call gendung\_init\_transparency

xor edx, edx ; size

mov ecx, esi ; file\_path

call engine\_mem\_load\_file

mov esi, eax

xor ecx, ecx

```

### About the SYM

The [diasurgical/scalpel](https://github.com/diasurgical/scalpel) repository includes a copy of a symbolic file that was

accidentally left on the Japanese release of Diablo on Playstation 1. The CD

contained debug information in a .SYM file, the format of which has been

reversed, so we can recover type information, variable names, etc, for the PSX

release.

\* Download and open [jap\_05291998.out](https://raw.githubusercontent.com/diasurgical/scalpel/master/psx/symbols/jap\_05291998.out)

\* Example: search for `LoadL1Dungeon\_\_FPcii`

\* Starting memory address `0x8013CF64`

\* Function name `LoadL1Dungeon`

\* Function arguments `(\*char sFilename, int vx, int, vy)`

\* #TODO what else can be inferred from below?

```

135ea8: $8013cf64 8c Function\_start

fp = 29

fsize = 48

retreg = 31

mask = $80070000

maskoffs = -4

line = 905

file = C:\diabpsx\SOURCE\DRLG\_L1.CPP

name = LoadL1Dungeon\_\_FPcii

135ef4: $00000010 94 Def class REGPARM type PTR CHAR size 0 name sFileName

135f0b: $00000011 94 Def class REGPARM type INT size 0 name vx

135f1b: $00000012 94 Def class REGPARM type INT size 0 name vy

135f2b: $8013cf64 90 Block\_start line = 1

135f34: $00000005 94 Def class REG type INT size 0 name i

135f43: $00000007 94 Def class REG type INT size 0 name j

135f52: $0000000b 94 Def class REG type INT size 0 name rw

135f62: $0000000c 94 Def class REG type INT size 0 name rh

135f72: $00000010 94 Def class REG type PTR UCHAR size 0 name pLevelMap

135f89: $00000008 94 Def class REG type PTR UCHAR size 0 name lm

135f99: $8013d0c4 90 Block\_start line = 44

135fa2: $8013d11c 92 Block\_end line = 60

135fab: $8013d11c 92 Block\_end line = 60

135fb4: $8013d138 8e Function\_end

```

## Comparing a function with the original exe

### Using Riivaaja

\* Step 1:

https://docs.docker.com/install/

\* Step 2:

Download latest devilution-comparer: https://github.com/diasurgical/devilution-comparer/releases (build from src if on Mac)

\* Step 3:

Get the Diablo 1.09 exe

\* Step 4:

If not on Windows Devilution-comparer requires Wine, either install Wine or use Riivaaja as a proxy (more on this later if you would like to go this route).

\* Step 5:

#### To get a function for comparison

Build:

`docker run --rm -v $(pwd):/root/devilution -e MAKE\_BUILD=pdb diasurgical/riivaaja`

Generate diff:

`devilution-comparer Diablo\_original.exe Diablo.exe <function\_name>`

You can add `--no-mem-disp` if you want a cleaner output but this can also hide valuable details

This will generate an `orig.asm` and `compare.asm` that you can compare in your favorit `diff` application, in the folder that you can the command from.

To use Riivaaja instead of installing Wine, create `wine` in your `$PATH` and add this content:

```bash

#!/bin/sh

docker run --rm -v $(pwd):/root/devilution --entrypoint "/usr/bin/wine" diasurgical/riivaaja:stable $(basename $1) $2 $3

```

(Don't forget to also set exec permissions on the file)

### Using devilution-comparer with Wine

Install dependencies:

1. Install Wine if not on Windows (e.g. `sudo pacman -S wine`)

2. Install MS VC+ 5 + SP3 and MS VC+ 6 + SP5 + PP. (for more information see the [building instructions](https://github.com/diasurgical/devil-nightly#building-with-visual-c-6) of the readme)

Install `devililution-comparer` from release (or from source below):

1. Download and extract the latest release from https://github.com/diasurgical/devilution-comparer/releases

Or install `devililution-comparer` from source:

1. `git clone https://github.com/diasurgical/devililution-comparer`

2. `cd devililution-comparer`

3. `cargo build --release`

4. `cp cvdump.exe target/release/`

5. `cp comparer-config.toml target/release/`

Clone Devilution nightly, build and compare against the original Diablo binary:

1. `git clone https://github.com/diasurgical/devil-nightly`

2. `make MAKE\_BUILD=pdb -f MakefileVC`

3. `cp /path/to/diablo-v1.09b.exe .`

4. `../devilution-comparer/target/debug/devilution-comparer diablo-v1.09b.exe Diablo.exe <function name>` (replace `<function name>` with e.g. `InitMonsterTRN`)

5. `code --diff orig.asm compare.asm` (or `diff -u orig.asm compare.asm`)

To watch build directory for changes use the `-w` command line flag:

```bash

$ ./devilution-comparer -w diablo-v1.09b.exe Diablo.exe InitMonsterTRN

Found InitMonsterTRN at 0x4322EC, size: 0x8C; orig size: 0x8C

Started watching Diablo.pdb for changes. CTRL+C to quit.

```