The PDB File Format

Introduction

PDB (Program Database) is a file format invented by Microsoft and which contains debug information that can be consumed by debuggers and other tools. Since officially supported APIs exist on Windows for querying debug information from PDBs even without the user understanding the internals of the file format, a large ecosystem of tools has been built for Windows to consume this format. In order for Clang to be able to generate programs that can interoperate with these tools, it is necessary for us to generate PDB files ourselves.

At the same time, LLVM has a long history of being able to cross-compile from any platform to any platform, and we wish for the same to be true here. So it is necessary for us to understand the PDB file format at the byte-level so that we can generate PDB files entirely on our own.

This manual describes what we know about the PDB file format today. The layout of the file, the various streams contained within, the format of individual records within, and more.

We would like to extend our heartfelt gratitude to Microsoft, without whom we would not be where we are today. Much of the knowledge contained within this manual was learned through reading code published by Microsoft on their GitHub repo.

File Layout

Important

Unless otherwise specified, all numeric values are encoded in little endian. If you see a type such as `uint16_t` or `uint64_t` going forward, always assume it is little endian!

The MSF Container

A PDB file is an MSF (Multi-Stream Format) file. An MSF file is a "file system within a file". It contains multiple streams (aka files) which can represent arbitrary data, and these streams are divided into blocks which may not necessarily be contiguously laid out within the MSF container file. Additionally, the MSF contains a stream directory (aka MFT) which describes how the streams (files) are laid out within the MSF.

For more information about the MSF container format, stream directory, and block layout, see The MSF File Format.

Streams

The PDB format contains a number of streams which describe various information such as the types, symbols, source files, and compilands (e.g. object files) of a program, as well as some additional streams containing hash tables that are used by debuggers and other tools to provide fast lookup of records and types by name, and various other information about how the program was compiled such as the specific toolchain used, and more. A summary of streams contained in a PDB file is as follows:

Name	Stream Index	Contents
Old Directory	• Fixed Stream Index 0	• Previous MSF Stream Directory
PDB Stream	• Fixed Stream Index 1	• Basic File Information • Fields to match EXE to this PDB • Map of named streams to stream indices
TPI Stream	• Fixed Stream Index 2	• CodeView Type Records • Index of TPI Hash Stream
DBI Stream	• Fixed Stream Index 3	• Module/Compiland Information • Indices of individual module streams • Indices of public / global streams • Section Contribution Information • Source File Information • References to streams containing FPO / PGO Data
IPI Stream	• Fixed Stream Index 4	• CodeView Type Records • Index of IPI Hash Stream
/LinkInfo	• Contained in PDB Stream Named Stream map	• Unknown
/src/headerblock	• Contained in PDB Stream Named Stream map	• Summary of embedded source file content (e.g. natvis files)
/names	• Contained in PDB Stream Named Stream map	• PDB-wide global string table used for string de-duplication
Module Info Stream	• Contained in DBI Stream • One for each compiland	• CodeView Symbol Records for this module • Line Number Information
Public Stream	• Contained in DBI Stream	• Public (Exported) Symbol Records • Index of Public Hash Stream
Global Stream	• Contained in DBI Stream	• Single combined symbol-table • Index of Global Hash Stream
TPI Hash Stream	• Contained in TPI Stream	• Hash table for looking up TPI records by name
IPI Hash Stream	• Contained in IPI Stream	• Hash table for looking up IPI records by name

More information about the structure of each of these can be found on the following pages:

The PDB Info Stream (aka the PDB Stream)

Information about the PDB Info Stream and how it is used to match PDBs to EXEs.

The PDB TPI and IPI Streams

Information about the TPI stream and the CodeView records contained within.

The PDB DBI (Debug Info) Stream

Information about the DBI stream and relevant substreams including the Module Substreams, source file information, and CodeView symbol records contained within.

The Module Information Stream

Information about the Module Information Stream, of which there is one for each compilation unit and the format of symbols contained within.

The PDB Public Symbol Stream

Information about the Public Symbol Stream.

The PDB Global Symbol Stream

Information about the Global Symbol Stream.

The PDB Serialized Hash Table Format

Information about the serialized hash table format used internally to represent things such as the Named Stream Map and the Hash Adjusters in the TPI/IPI Stream.

CodeView

CodeView is another format which comes into the picture. While MSF defines the structure of the overall file, and PDB defines the set of streams that appear within the MSF file and the format of those streams, CodeView defines the format of **symbol and type records** that appear within specific streams. Refer to the pages on CodeView Symbol Records and CodeView Type Records for more information about the CodeView format.