

---

# PDT's ductape API Tutorial

## Table of Contents

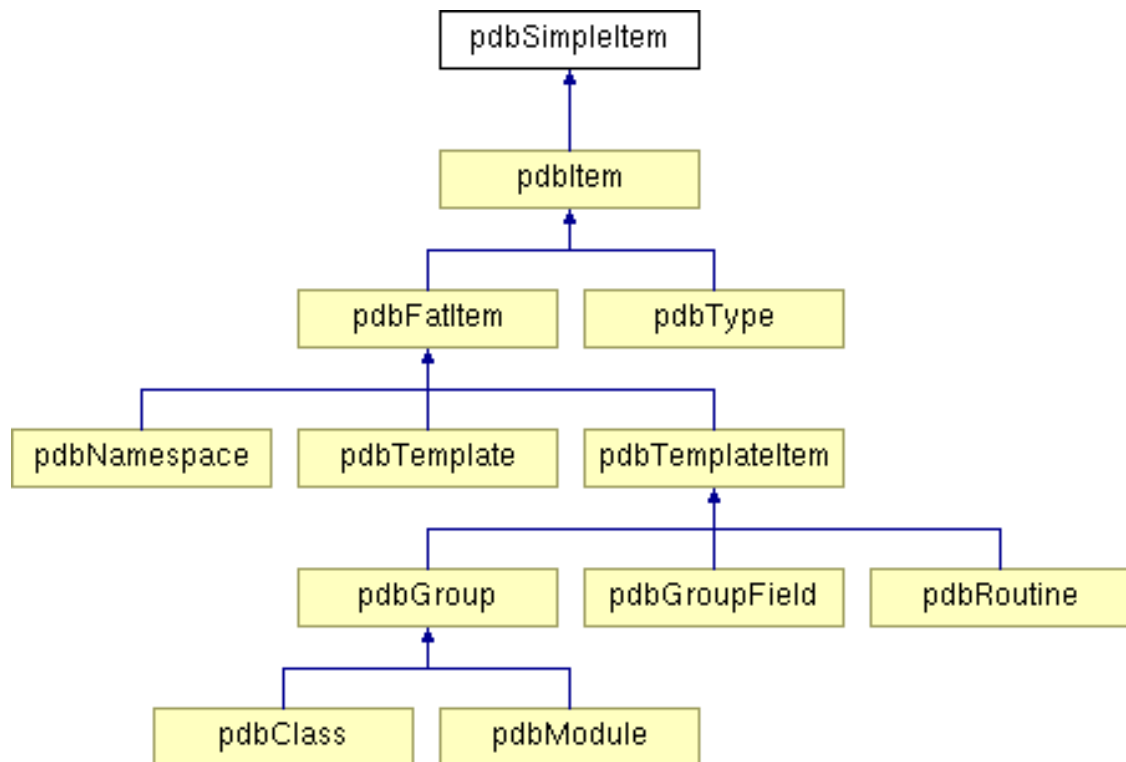
1. Overview of PDT's Architecture .....	1
2. Using Iterators .....	1

## 1. Overview of PDT's Architecture

In this tutorial you will learn how to use PDT's ductape API to read PDB files. A PDB file is created by parsing a C/C++ or Fortran source with the include parser: `cparse`, `cxxparse`, `f90parse` and `f95parse`. To use these parsers type: `%> export PATH=[path-to-pdt]/[arch]/bin/ %> cxxparse program.cpp %> ls program.ccp program.pdb`

The Ductape API is organized as a hierarchy of classes. Here is a picture represent some of those classes. More detail can be found at API documentation [<http://www.cs.uoregon.edu/research/pdt/docs/ductape/index.html>]

**Figure 1. Partial hierarchy of the classes in the ductape API**



## 2. Using Iterators

Ductape API allows iterators to be used in many of its classes. Using iterators will allow you to access member of a datastructure without needing to know the underlying implementation.

One place where you can use iterators is in the `pdb` class, here is a simple function that iterates over every class in the file printing its name. File to be parsed and analyzed: `class bar { int foo(int v) { return v + 2; } class bar2 { int routine(bool t) {return 0;} }; int a; }; C source file: #include "pdbAll.h" #include "stdio.h" int main(int argc, char *argv[]) { // Read the pdb file as input for this program. PDB p(argv[1]); if ( !p ) return 1; // Iterate through each class in the pdb file and print its name. for (PDB::classvec::iterator r = p.getClassVec().begin(); r!=p.getClassVec().end(); r++) { cout << (*r)->name() << endl; } return 0; } To run type: %> g++ -I../inc/ -o vector vector.cc ../lib/libpdb.a %> cxxparse testApp.cc %> ./vector testApp.pdb bar bar2 There is a collection of example source code in the API documentation [http://www.cs.uoregon.edu/research/pdt/docs/ductape/examples.html].`