Welcome

db4o is the native Java, .NET and Mono open source object database.

This documentation and tutorial is intended to get you started with db4o and to be a reliable companion while you develop with db4o. Before you start, please make sure that you have downloaded the latest db4o distribution from the db4objects website.

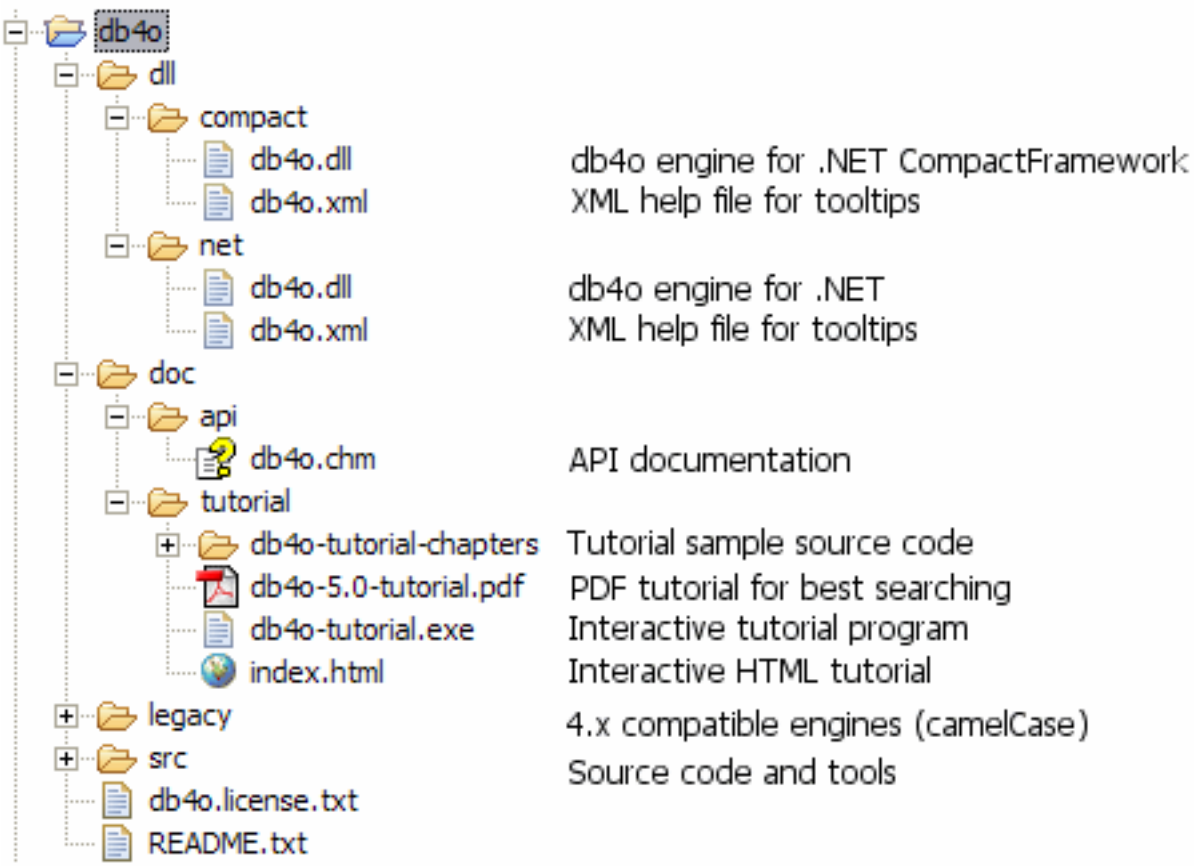
You are invited to join the db4o community in the public db4o forums to ask for help at any time. You may also find the db4o knowledgebase helpful for keyword searches.

Java, .NET and Mono

db4o is available for Java, for .NET and for Mono. This tutorial was written for .NET . The structure of the other distributions may be considerably different, so please use the tutorial for the version that you plan to experiment with first.

Download Contents

The db4o .NET distribution comes as one MSI installer file, db4o-5.0-net.msi. After you run the installer, you get the following directory structure:



db4o-5.0/doc/tutorial/db4o-tutorial.exe

This is the interactive tutorial application for .NET. Examples can be run "live" against a db4o database from within the application.

db4o-5.0/doc/tutorial/db4o-5.0-tutorial.pdf

The PDF version of the tutorial allows best fulltext search capabilities.

db4o-5.0/doc/api/db4o.chm

The API documentation for db4o is supplied as a compiled Help file. While you read through the rest of this tutorial, it may be helpful to look into the API documentation occasionaly.

1. First Glance

Before diving straight into the first source code samples let's get you familiar with some basics.

1.1. The db4o engine...

The db4o object database engine consists of one single DLL. This is all that you need to program against. The versions supplied with the distribution can be found in /db4o-5.0/dll/.

db4o is available in two seperate distributions for Microsoft .NET. One distribution is for the .NET Framework 1.0/1.1 and the other is for the .NET Framework 2.0. Be sure to use the correct one for your project environment.

/db4o-5.0/dll/net/db4o.dll
is the standard db4o engine for the .NET framework.

/db4o-5.0/dll/compact/db4o.dll
is built for the .NET CompactFramework.

1.2. Installation

To use db4o in a development project, you only need to add one of the above db4o.dll files to your project references.

Here is how to do this with Visual Studio .NET:
- copy db4o.dll to your VS.NET project folder
- Right-click on "References" in the Solution Explorer
- choose "Add Reference"
- select "Browse"
- find the db4o.*dll in your project folder
- click "Open"
- click "OK"
- Right-click db4o
- Choose "Properties"
- Set "Copy Local" to [True] if it is not already set.

1.3. db4o Object Manager

db4o Object Manager is a GUI tool to browse and query the contents of any db4o database file. Object Manager has to be downloaded seperately from the main db4o distributions. Please visit the db4o Download Center and choose the installation appropriate for your system. The following distributions are currently available:

- db4o ObjectManager for Windows IKVM (Java VM included)
- db4o ObjectManager for Windows no Java VM
- db4o ObjectManager for Linux

1.4. API Overview

Do not forget the API documentation while reading through this tutorial. It provides an organized view of the API, looking from a namespace perspective and you may find related functionality to the theme you are currently reading up on.

For starters, the namespace com.db4o and com.db4o.query are all that you need to worry about.

com.db4o

The com.db4o namespace contains almost all of the functionality you will commonly need when using db4o. Two objects of note are com.db4o.Db4o, and the com.db4o.ObjectContainer interface.

The com.db4o.Db4o factory is your starting point. Static methods in this class allow you to open a database file, start a server, or connect to an existing server. It also lets you configure the db4o environment before opening a database.

The most important interface, and the one that you will be using 99% of the time is com.db4o.ObjectContainer: This is your db4o database.
- An ObjectContainer can either be a database in single-user mode or a client connection to a db4o server.
- Every ObjectContainer owns one transaction. All work is transactional. When you open an ObjectContainer, you are in a transaction, when you commit() or rollback(), the next transaction is started immediately.
- Every ObjectContainer maintains it's own references to stored and instantiated objects. In doing so, it manages object identities, and is able to achieve a high level of performance.
- ObjectContainers are intended to be kept open as long as you work against them. When you close an ObjectContainer, all database references to objects in RAM will be discarded.

com.db4o.ext

In case you wonder why you only see very few methods in an ObjectContainer, here is why: The db4o interface is supplied in two steps in two namespaces , com.db4o and com.db4o.ext for the following reasons:
- It's easier to get started, because the important methods are emphasized.
- It will be easier for other products to copy the basic db4o interface.
- It is an example of how a lightweight version of db4o could look.

Every com.db4o.ObjectContainer object is also an com.db4o.ext.ExtObjectContainer. You can cast it to ExtObjectContainer or you can use the to get to the advanced features.

com.db4o.config