HammerDB v2.14 Installation and Troubleshooting Guide

This guide gives you an introduction to installing and troubleshooting HammerDB v2.14 on Linux and Windows and compiling for UNIX platforms. Please read this guide carefully before raising support issues.

Hammerora to HammerDB Renaming

Hammerora has been renamed to HammerDB in recognition of the wider number of databases supported since the software's first release. Whilst documentation is in progress existing document references to Hammerora v2.11 and earlier can be used interchangeably with HammerDB with regards to functionality.

HammerDB v2.14 New Features

In HammerDB there are updated binaries in the precompiled packages to take advantage of the new features of TCL 8.6 in future releases. All database packages have been recompiled to versions:

tcl8.6, tk8.6, thread2.7.0 (included with tcl8.6), oratcl4.5, mysqltcl3.052 pgtcl2.0.0 tclodbc2.5.1

All packages have been tested and verified against Linux x86 and x86-64 and Windows x86 and x86-64

Bug Fixes

[#93] Oracle PL/SQL Schema Build Fails on Index Build
[#94] SQL Server Transaction Counter reports Negative Number then Zero*
[#95] pgsqlXXX is not a valid postgresql connection

*SQL Server Users should note that due bug #94 and an issue in SQL Server the transaction rate used for TPM has changed from previous versions to Batches/Sec. This new value also coincides with the transaction rate reported by SQL Server Management Studio.

Removed tclodbc builds on Linux releases and greyed out SQL Server benchmark option to make tclodbc and SQL Server Windows only options

Added Order Line Tablespace option to Oracle Database TPC-C schema to enable partitioned Order Line table to be placed in a separate cache with a different block size from the standard cache.

Implemented TimesTen TPC-C Partitioning Option

Implemented TimesTen TPC-H Workload

Added SQL Server TPC-C workload from schema and Stored Procedures provided by Thomas Kejser under "updated" option. Regular schema remains available under "original" option

---

Tested Platforms and Databases

HammerDB has been built and tested on the following 32 and 64-bit Linux and Windows platforms.

Linux	openSUSE 12.2 openSUSE 12.3 SUSE Linux Enterprise 11 SP2 Red Hat/Oracle Linux 6.2 Red Hat/Oracle Linux 6.3
Windows	Windows 7 Windows Server 2008 Windows Server 2012
Databases	SQL Server 2008 SQL Server 2012* Oracle 9i Oracle 10g Oracle 11g Oracle 12c Oracle TimesTen 11.2.2.4.1** MySQL 5.5 MySQL 5.6 PostgreSQL 9.1 PostgresQL 9.2

	EnterpriseDB Postgres Plus Advanced Server 9.0*** EnterpriseDB Postgres Plus Advanced Server 9.1*** EnterpriseDB Postgres Plus Advanced Server 9.2*** Greenplum Database Community Edition 4.2.1.0 **** Redis 2.6.13

*HammerDB has not been tested on any version of Microsoft SQL Server prior to 2008 and no updates will made to support prior releases. In particular functionality used in HammerDB is known absolutely to not be compatible with SQL Server 2000 or earlier. SQL Server 2005 has not been tested. For SQL Server 2012 use SQL Server Native Client 11.0, for SQL Server 20008 use SQL Server Native Client 10.0.

**HammerDB has been verified against TimesTen version 11.2.2.4.1, HammerDB is known not to work against TimesTen version 11.2.2.4.0 or any earlier version and no updates will made to support prior releases.

***PostgreSQL support has been provided both for native PostgreSQL and EnterpriseDB's Oracle compatible mode. Oracle compatible mode will only work against EnterpriseDB Postgres Plus Advanced Server. With Oracle compatible mode DRITA functionality does not work against the initial release of Postgres Plus Advanced Server 9.1AS due to a bug within the database software. To enable DRITA functionality apply a patch to resolve this issue and test to ensure that DRITA snapshots work.

****The Greenplum Database is supported for the PostgreSQL TPC-H workload only.

You may compile from source to support non Linux or Windows platforms however none of these platforms have been tested and for performance reasons critical to load testing installing HammerDB on Linux or Windows on Intel x86/x86-64 based architectures is strongly recommended even for testing databases on different platforms.

Understanding 32 and 64-bit installations

From version 2.12 HammerDB is available in a pre-compiled packaged format for 32-bit/x86 Linux, 64-bit/x86-64 Linux, 32-bit/x86 Windows and 64-bit/x86-64 Windows. The most common HammerDB installation error results from mixing incompatible 32 and 64-bit software installations. A 32-bit version of HammerDB requires a 32-bit operating system and the 32-bit client libraries for your chosen database, similarly a 64-bit version of HammerDB requires a 64-bit operating system and the 64-bit client libraries for your chosen database. In some instances a 64-bit operating system may support a 32-bit version of HammerDB, in this instance the 32-bit database client libraries are required. You should consult your database vendors support matrix for guidance on whether running 32-bit database client software on a 64-bit operating system is supported. Note this guidance applies to the client software used by HammerDB only, the Database server itself if running on another server is independent and mixing 32 and 64-bit clients and servers and different operating systems is usually a supported configuration. The most significant difference between the 32-bit and 64-bit versions of HammerDB is that the 2GB memory limitation is removed with the 64-bit installation, consequently the 64-bit version of HammerDB will support the creation of a significantly greater number of virtual users (subject to RAM availability) than the 32-bit version. Use the following instructions to determine whether your system is 32 or 64-bit for Linux and Windows.
Carefully ensuring that you have installed a full stack of 32-bit or 64-bit operating system, database client and HammerDB can prevent your repeating most common installation errors experienced.

Linux

On a Linux operating system run the command ***uname –m*** to determine whether your installation is 32 or 64-bit. If the command returns the following:

```
[root@system1 ~]# uname -m

i686
```

Your system is 32-bit and you need the x86 installer package for HammerDB. If however the result is as follows

```
[root@system2 ~]# uname -m

x86_64
```

Your system is 64-bit and you need the x86-64 installer package.

If you have a system with HammerDB pre-installed to check the installation look in the `readme` file. For example the following line:

```
HammerDB v2.12 for Linux 64-bit x86-64
```

denotes a 64-bit installation. You can verify this by running the following command in the HammerDB console to show the machine type which in this case is 64-bit (note AMD64 denotes the architecture and will be the output on both Intel and AMD processors):

```
(HammerDB-2.12) 28 % puts $tcl_platform(machine)
amd64
```

You can also use the Linux `file` command on the executable tclsh8.6 and wish8.6 in the bin directory to verify the compatibility.

Windows

Microsoft has published an FAQ on 32-bit and 64-bit installations here.

32-bit-and-64-bit-Windows-frequently-asked-questions

As detailed in this FAQ right-clicking on Computer and selecting properties displays the system type. The following example shows a 64-bit x86-64 Windows installation.

Figure 1 64-bit Windows

Windows users should also be aware that Windows enables the emulation of 32-bit Windows environment on a 64-bit operating system with an implementation called WOW64 . Windows users should therefore pay particular attention to the 32 and 64-bit installations of their environment. For example a 32-bit database client may be installed without error on 64-bit Windows but HammerDB for 64-bit Windows will not be able to load the 32-bit client libraries. You should not run performance critical tests under the 32-bit emulated environment and should instead use the 64-bit native version of HammerDB and the accompanying 64-bit database libraries. However for educational purposes where only a 32-bit database is available such as the Express Editions of Oracle and SQL Server HammerDB for 32-bit will provide full functionality against this

environment on a 64-bit Windows installation.

When installing HammerDB on a 64-bit Windows environment the installer will automatically detect your environment and recommend the installation of HammerDB in the correct location. For the 64-bit application this will be in `C:\Program Files` and for the 32-bit application this will be in `C:\Program Files (x86)` as shown in figure 2.

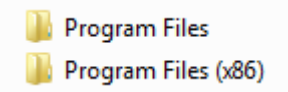

Figure 2 Program Files

Before installing HammerDB on Windows Oracle users of 32-bit software on 64-bit operating systems should be aware of Oracle bug #3807408.
This bug in some versions of the Oracle client and database software causes Oracle error:

ORA-12154: TNS:could not resolve the connect identifier specified

This bug is caused whenever any Oracle client program (including HammerDB) is installed in a directory containing parenthesis such as the following:

"C:\Program Files (x86)\..."

(NOTE: This is an Oracle software bug not a HammerDB one).

The Oracle recommended workaround is as follows:

Use a version of the Oracle client AND database software that contains the fix for Bug 3807408. This fix requires that both the client and database software be patched.

OR

Find the location of the application that is generating the error. Check the path to this location and see if it contains any parenthesis. If so, you must relocate the application to a directory without any parenthesis in the path.

Therefore if running HammerDB on Windows and your client or database is affected by Oracle bug 3807408 then either patch Oracle or ensure that HammerDB is installed to a directory that does not contain parenthesis.

Verifying the Database Client Libraries

For Oracle SQL Server, MySQL and PostgreSQL it is essential that HammerDB can find the correct client libraries on the HammerDB server to connect to the chosen database. In most cases this means following the installation instructions for a client or server installation of the database software for Oracle, SQL Server, MySQL or PostgreSQL respectively.

HammerDB will not be able to connect to Oracle, SQL Server, MySQL or PostgreSQL if you have not installed and correctly configured the database client libraries.

For Redis the client interface is included and no additional client libraries are required.

Oracle

For Oracle on Linux the important environment variables are ORACLE_HOME and LD_LIBRARY_PATH and on Windows the PATH environment variable is set automatically during installation in the system properties. When using the Oracle instant client Oratcl uses the additional environment variable ORACLE_LIBRARY to identify the Oracle client library. On the Windows the Oracle client library is called oci.dll in a location such as:

```
C:\oraclexe\app\oracle\product\11.2.0\server\bin
```

On Linux the library is called libclntsh.so where this is typically a symbolic link to a product specific name such as libclntsh.so.11.1 for Oracle 11g.

An example .bash_profile file is shown for a typical Oracle environment.

```
oracle@server1  oracle]$ cat ~/.bash_profile
# .bash_profile

if [ -t 0 ]; then
stty intr ^C
fi

# Get the aliases and functions
if [ -f ~/.bashrc ]; then
        . ~/.bashrc
fi
# User specific environment and startup programs
umask 022
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/11.2.0/dbhome_1
export LD_LIBRARY_PATH=$ORACLE_HOME/lib
export ORACLE_LIBRARY=$ORACLE_HOME/lib/libclntsh.so
export ORACLE_SID=PROD1
export PATH=$ORACLE_HOME/bin:$PATH
```

For Oracle TimesTen configuration see the supplementary HammerDB TimesTen OLTP guide. In particular note that although TimesTen uses OCI connectivity all required Oracle client software should be installed from the TimesTen installation software and not from a standard Oracle install.

SQL Server

On SQL Server on Windows the client libraries and necessary environment variables are set automatically during the SQL Server installation. Note that on 64-bit Windows the 64-bit ODBC client library is named ODBC32.DLL in the following location. C:\Windows\System32\odbc32.dll and the 32-bit library is in C:\Windows\SysWOW64\odbc32.dll.

MySQL

HammerDB version 2.14 has been built and tested against a MySQL 5.6 client installation. On Linux this means that HammerDB will require a MySQL client library called libmysqlclient.so.18. This client library needs to be referenced in the LD_LIBRARY_PATH in the same way described for Oracle previously in this section. If you do not have the correct client library you can either install a more recent client or create a

symbolic link named libmysqlclient.so.18 to an older library such as libmysqlclient.so.16. On Windows the MySQL client library is included with HammerDB.

PostgreSQL

For PostgreSQL the client library is called libpq.dll on Windows and libpq.so on Linux however note that additional libraries are also required. For Windows this means setting your PATH environment variable such as the following:

```
D:\PostgreSQL\pgsql\bin;
```

On Linux it is required to set the LD_LIBRARY_PATH environment variable in the same way described for Oracle previously in this section to the location of the PostgreSQL lib directory. Alternatively for installations of EnterpriseDB the client directory also contains the necessary files for a HammerDB installation. Where both 32 and 64-bit installations PostgreSQL is particularly sensitive to requiring a PATH or LD_LIBRARY_PATH with the required versions of the libraries. Failure to do this will result in the error "invalid argument" as shown.

```
% package require Pgtcl
couldn't load library "C:/Program Files/HammerDB-
2.12/lib/pgtcl2.0.0/libpgtcl.dll": invalid argument
```

When following the troubleshooting section further in this document to manually test PostgreSQL library loading if the error "invalid argument" is received ensure that the PATH or LD_LIBRARY_PATH environment variable points to the correct location only (and no other location). On Windows you also need to reboot your system when you have changed the PATH to ensure that the correct library is loaded without error.

Redis

The Redis client package is included with HammerDB for all installations and requires no further configuration.

Installing HammerDB on Linux and Windows

HammerDB is installed on both Linux and Windows with a graphical installer, consequently on Linux you need an X Windows environment with which to display the installer. Configure your DISPLAY environment variable to point to the correct X Windows display.

```
[oracle@server1 ~]$ export DISPLAY=server1:0.0
```

To start the installer on Linux make the installer file executable and then run the installer executable.

```
[oracle@server1 ~]$ chmod u+x HammerDB-2.14-Linux-x86-64-Install

[oracle@server1 ~]$ ./HammerDB-2.14-Linux-x86-64-Install
```

On Windows double-click on the setup file that is appropriate for your system.

Figure 3 HammerDB Windows Setup

The installer will start giving you the option of selecting the installation language

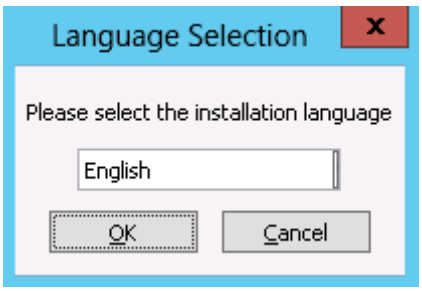

Figure 4 Select Language

You can then choose whether to continue with the installation

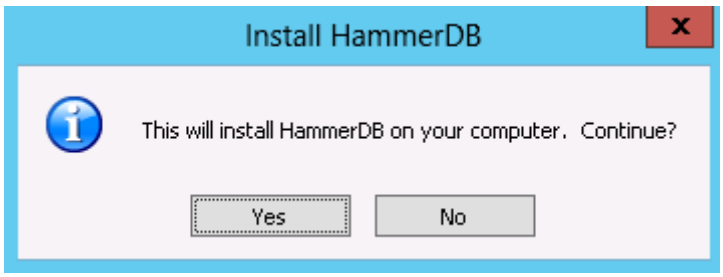

Figure 5 Continue

At the start of the install wizard, click next

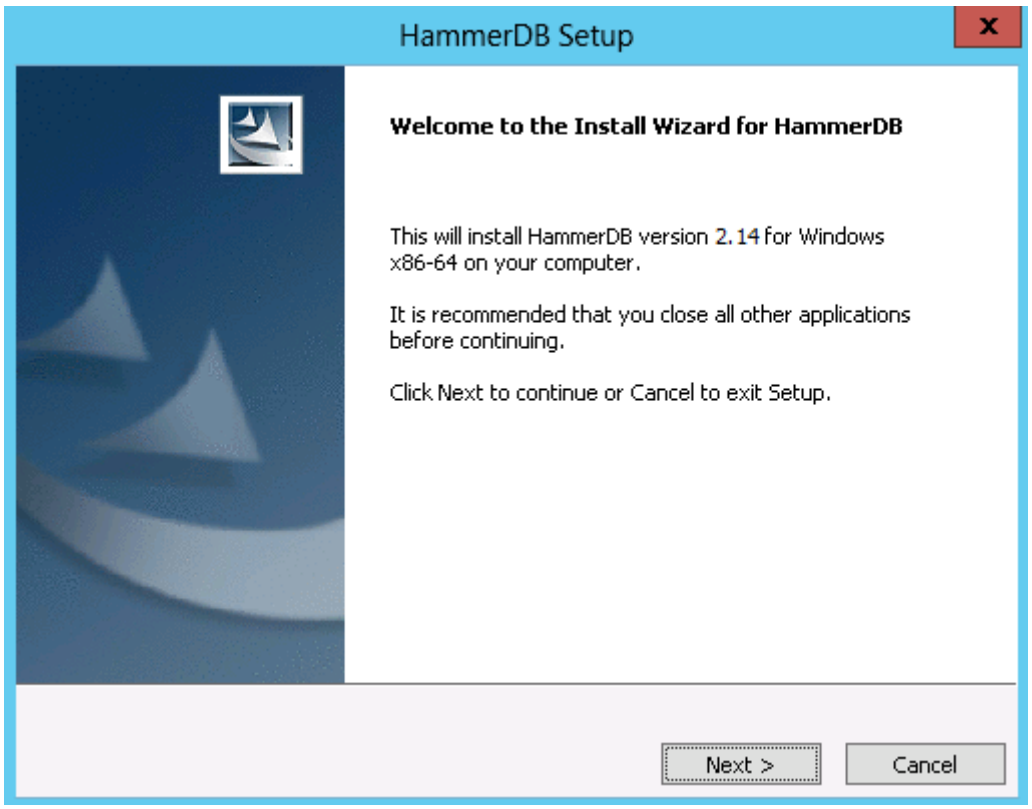

Figure 6 Welcome

Choose the destination location and Click Next. To change the default location Click Browse and select a new location.

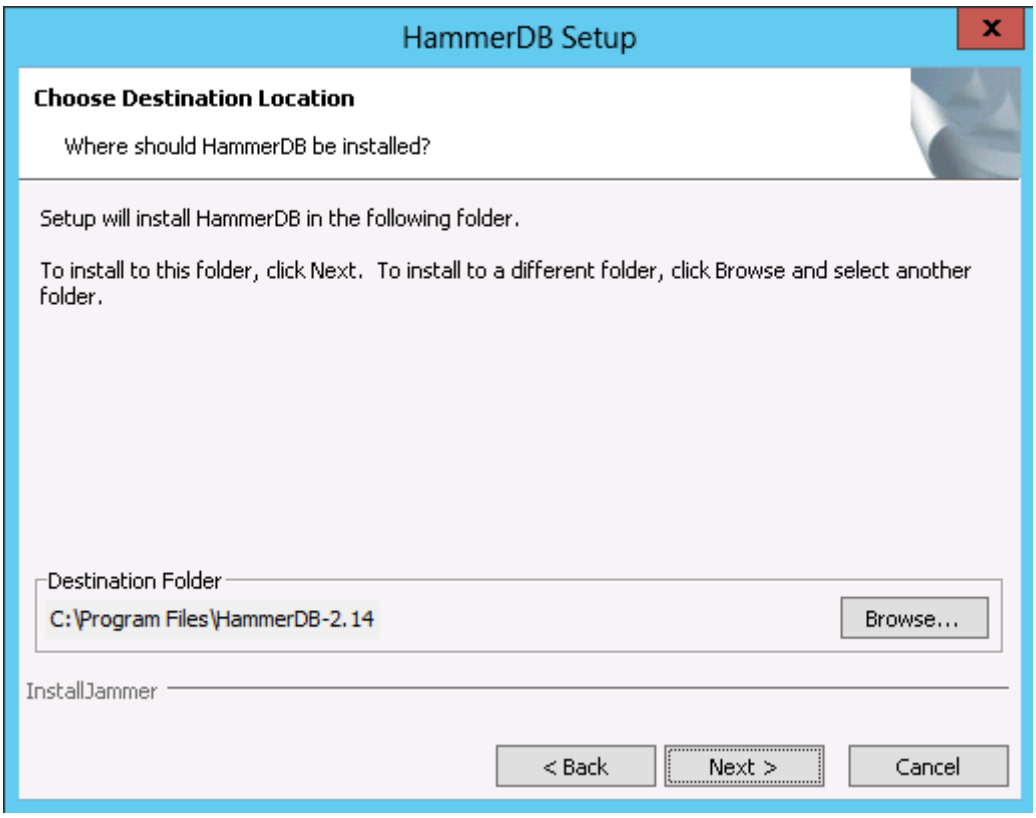

Figure 7 Choose Location

Click next to start copying the installation files

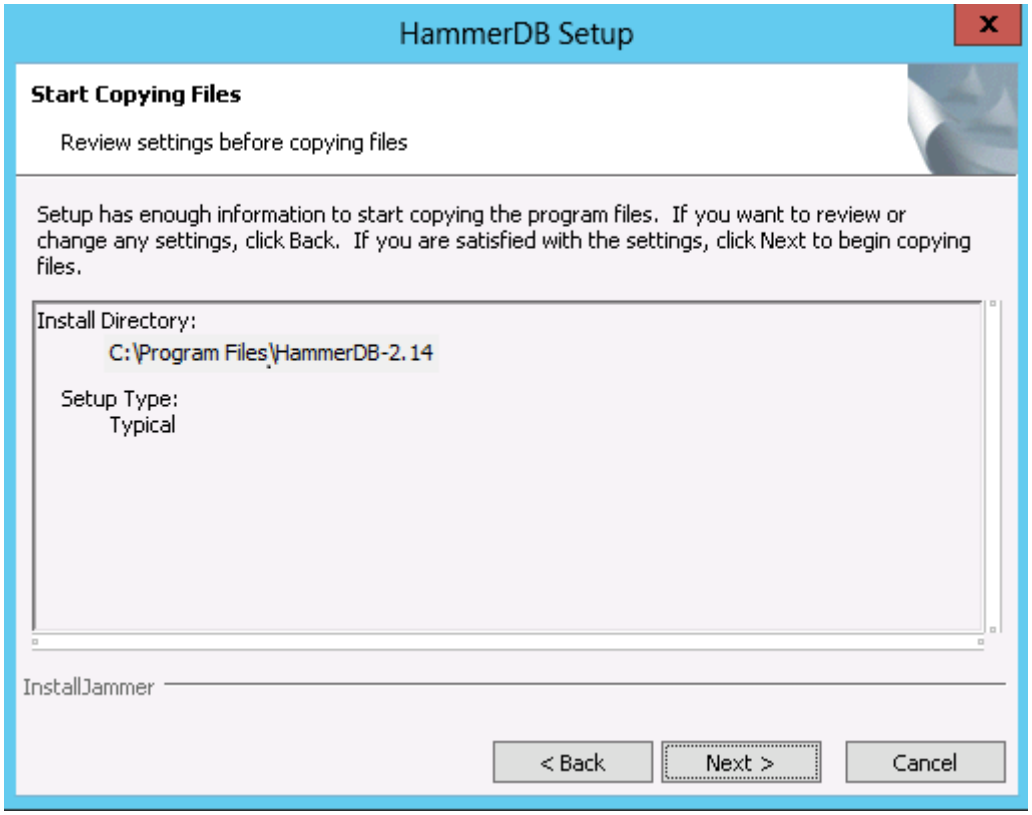

Figure 8 Start Copying

HammerDB will be installed in your selected location. On the completion screen click finish

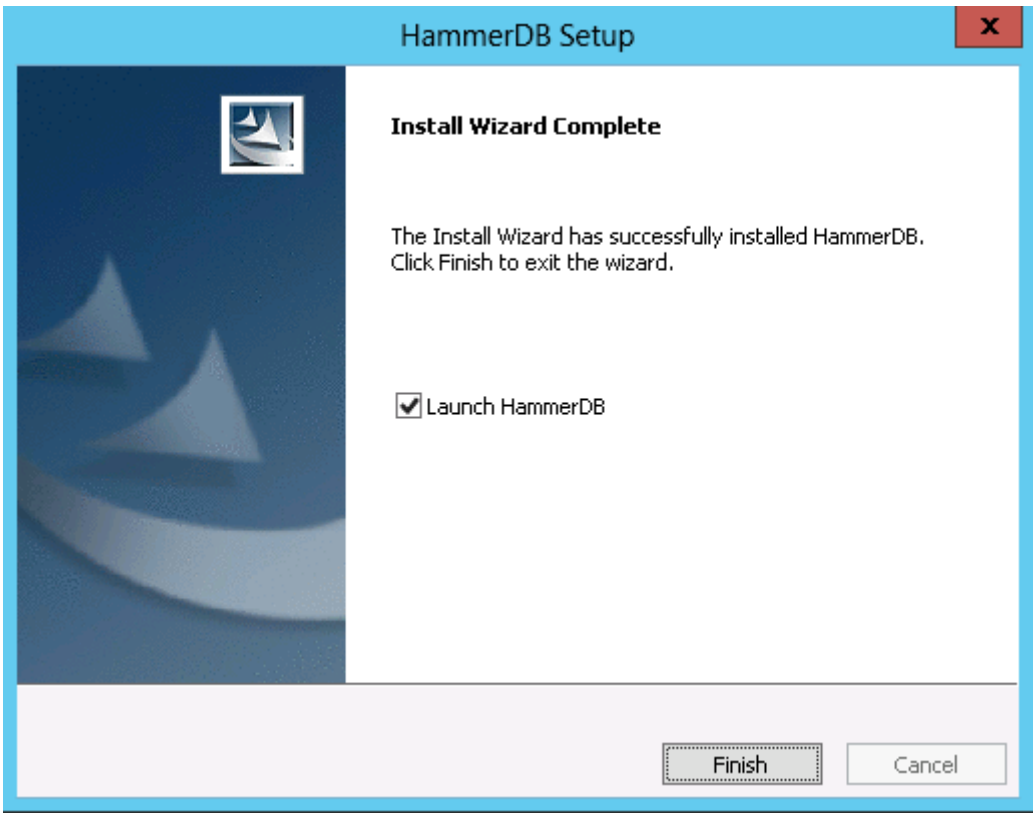

Figure 9 Complete

If you opt to launch HammerDB the main application window is displayed.

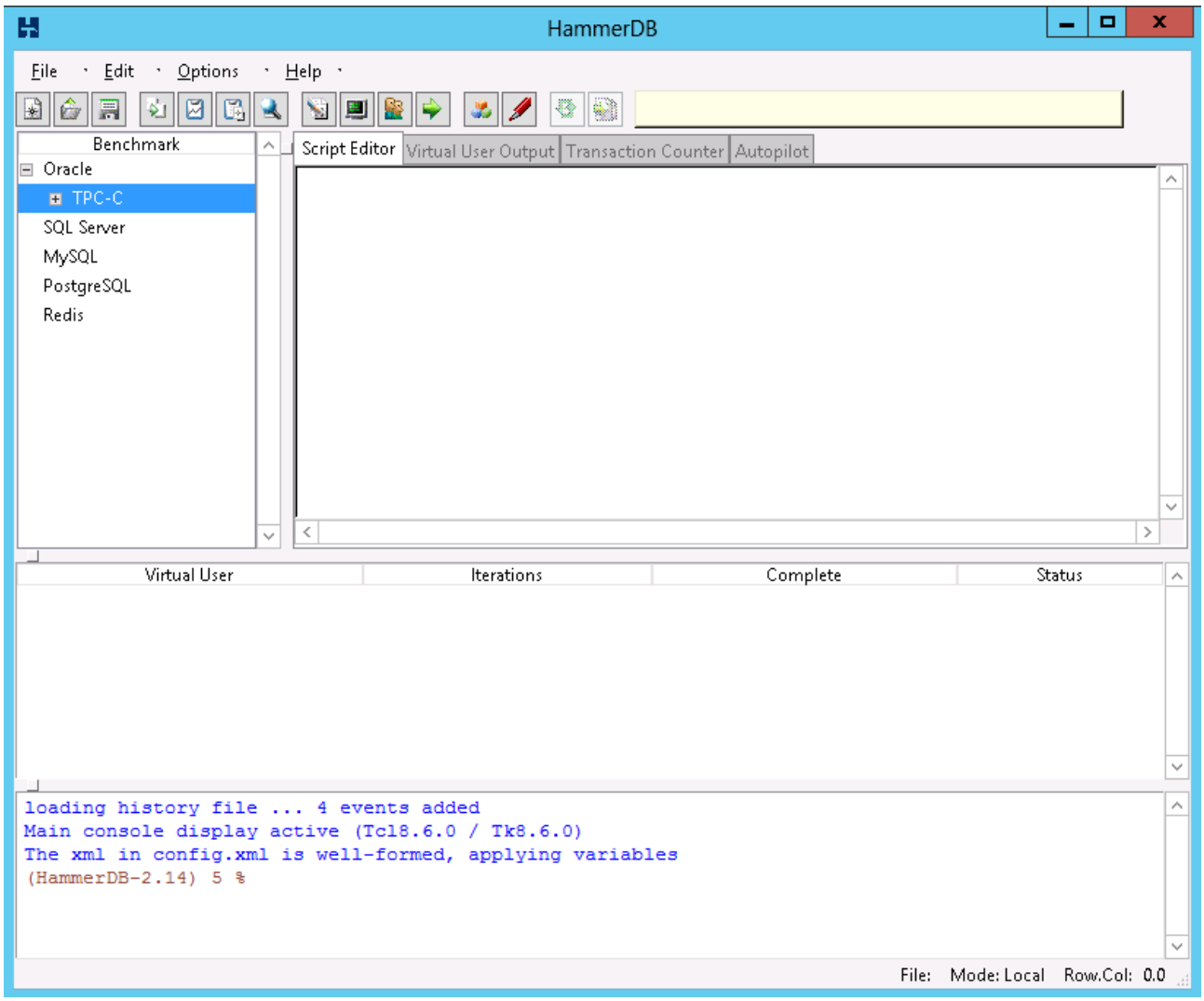

Figure 10 Application Window

Starting HammerDB

This section details how to start HammerDB after the software has been installed.

Linux

On Linux run HammerDB with the user with the correct permissions to access the database environment to which you are connecting. The following example shows the oracle user for which we have configured the Oracle client environment as shown previously in this document. Then set the DISPLAY environment variable.

```
[oracle@server1 HammerDB-2.12]$ export DISPLAY=server1:0.0
```

and as the oracle user run HammerDB.tcl directly from within the HammerDB directory.

```
[oracle@server1 HammerDB-2.14]$ ./hammerdb.tcl
```

Windows

To run HammerDB on Windows make sure that you have the correct permissions for your user to load the required database libraries. If you are unsure review the section on Microsoft UAC later in this document. Start HammerDB by double-clicking on the file `hammerdb.bat`.

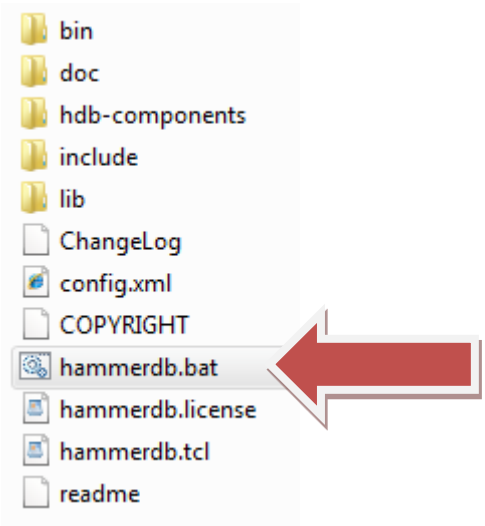

Figure 11 Starting HammerDB on Windows

Config.xml

In the HammerDB directory is a configuration file called config.xml that is read when HammerDB starts. If the config.xml file is correctly formed the data defined will be applied to the menus and options within HammerDB to enable customisation to your own environment.

At the top of the configuration file the value of the rdbms setting can be Oracle, MySQL, MSSQLServer , PostgreSQL or Redis

```
<?xml version="1.0" encoding="utf-8"?>
<hammerdb>
   <rdbms>Oracle</rdbms>
…
</hammerdb>
```

For the other config.xml values please consult the relevant guides for the workloads they represent.

Uninstalling HammerDB on Linux and Windows

HammerDB is entirely self-contained meaning that all the files installed are located in the HammerDB directory, no files or libraries are installed in any other location by HammerDB although HammerDB of course does require the pre-installation of the required client library files. To uninstall HammerDB on Linux run the uninstall executable as follows:

```
[oracle@server1 HammerDB-2.14]$ ./uninstall
```

On Windows double-click on the uninstall icon as shown.

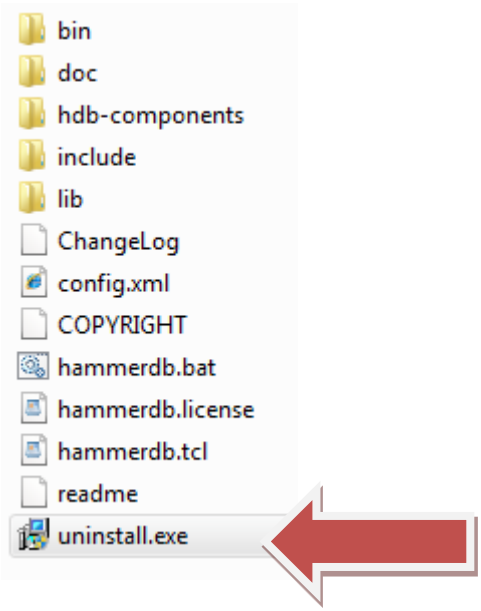

Figure 12 Uninstall.exe

Answer Yes to the prompt

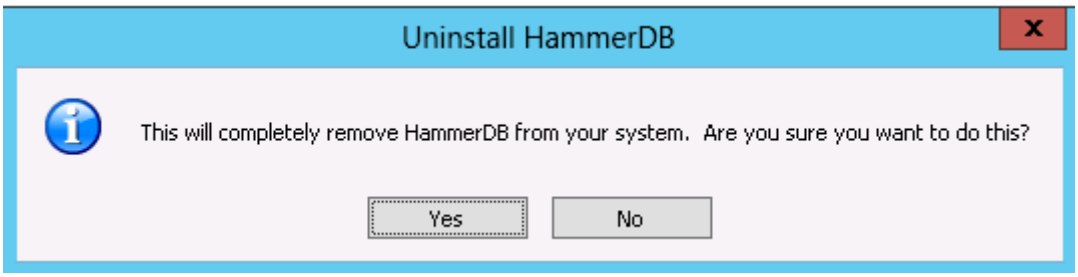

Figure 13 Uninstall Prompt

And HammerDB will be deleted from the system. Click Finish to complete the uninstall process.

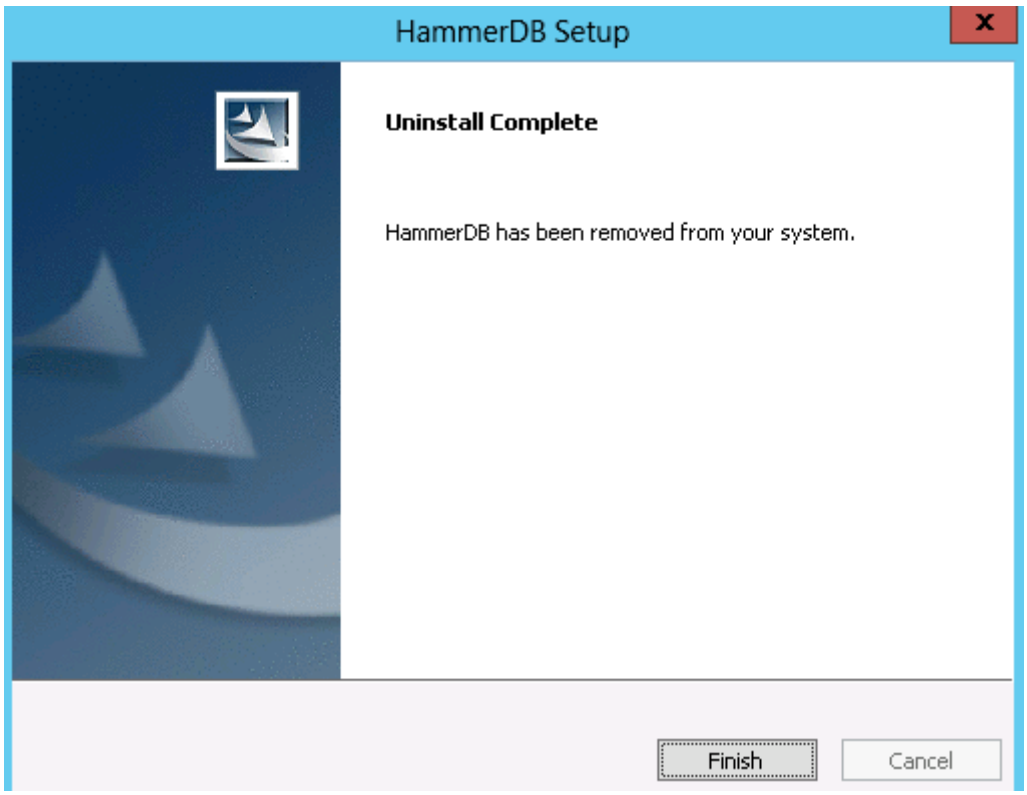

Figure 14 Uninstall Complete

Troubleshooting

The following sections give examples of how to troubleshoot installation and configuration issues with HammerDB.

Troubleshooting Library Errors

By far the most common HammerDB configuration error is to incorrectly setup the client libraries required for Oracle, SQL Server, MySQL and/or PostgreSQL respectively or to incorrectly mix 32 and 64-bit environments on the same system. The following example shows a library error reported by a virtual user when loading Oracle.

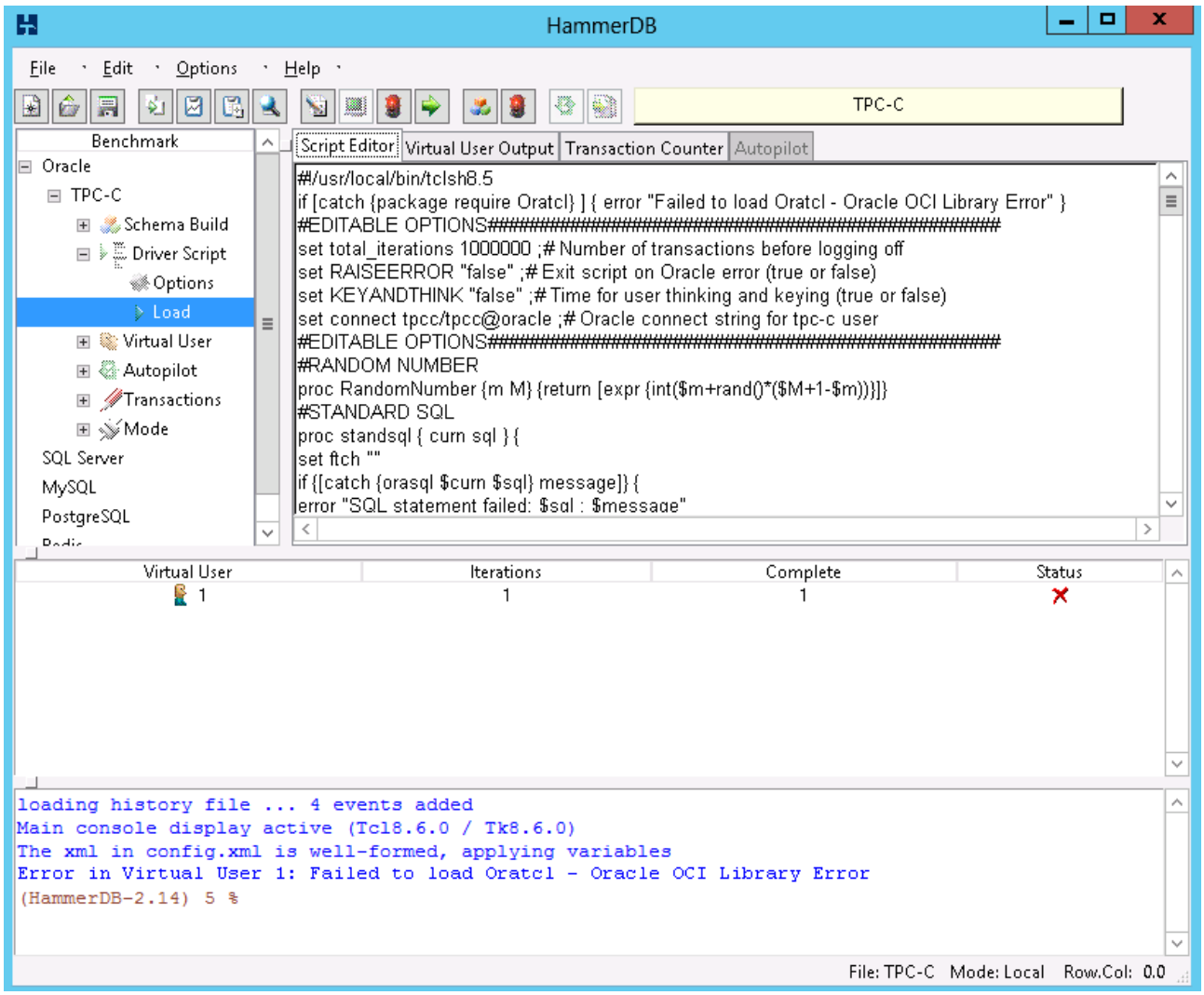

Figure 15 Oracle OCI Library Error

To troubleshoot library errors you can manually verify the library loading by starting the tclsh prompt and loading a library at the command line. On Linux for example you can do this by changing to the HammerDB directory, exporting the LD_LIBRARY_PATH environment variable to include the HammerDB lib directory to be searched first and then starting tclsh8.5 as follows:

```
[oracle@server1 HammerDB-2.14]$ export LD_LIBRARY_PATH=./lib:$LD_LIBRARY_PATH
```

The output shows that tclodbc has been successfully loaded.

```
[oracle@server1 HammerDB-2.14]$ ./bin/tclsh8.6
% package require tclodbc
2.5.1
```

On Windows double click on the bin directory as shown.

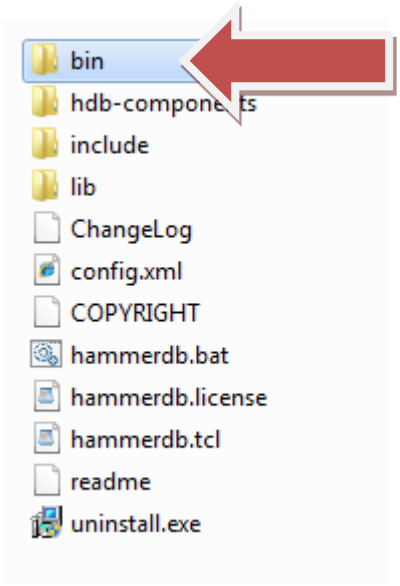

Figure 16 bin directory

Then double-click on tclsh86t.exe

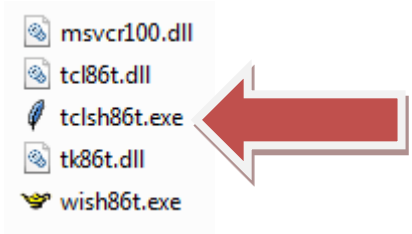

Figure 17 tclsh86t.exe

At the prompt type the command to load the library that has produced the error. The following example shows that the loading of Oratcl was not successful as oci.dll could not be loaded with error 126. This particular error means that the library cannot be found. Another common error, error 193 is the result of attempting to load the 64-bit oci.dll into the 32-bit version of HammerDB on Windows and is resolved by ensuring the correct 32-bit Oracle client is installed. Troubleshoot your error with the appropriate documentation for your database.

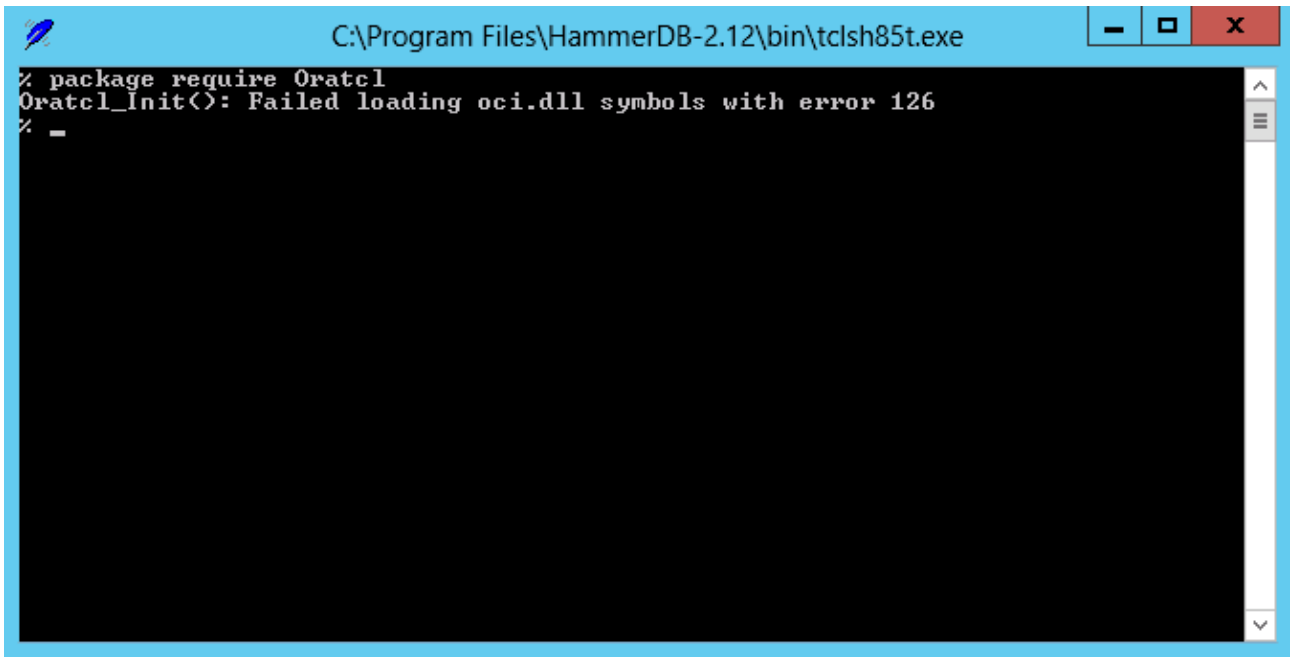

Figure 18 oci error 126

Version `GLIBC_2.7' not found

If on Linux platforms when running HammerDB you see an error such as version `GLIBC_2.17' not found and also find attempting to run the tclhsh8.6 or wish8.6 executables as shown below that you see the same error, this means that there is an incompatibility between the version of GLIBC that the precompiled version of HammerDB was compiled against and the version that you are running it on. GLIBC is forward compatible so HammerDB will run on more up to date platforms than the compiled platforms but not systems with older versions of GLIBC. Compilation and testing has taken place on Red Hat/Oracle Linux 6. Using this release or a more recent OS will typically resolve this issue.

```
$ wish8.6: /lib64/libc.so.6: version `GLIBC_2.7' not found (required by
./lib/libtk8.6.so)
wish8.6: /lib64/libc.so.6: version `GLIBC_2.7' not found (required by
./lib/libtcl8.6.so)
wish8.6: /lib64/libc.so.6: version `GLIBC_2.11' not found (required by
./lib/libtcl8.6.so)
```

You can check your version of GLIBC by running the following command, so for example this version of glibc from is compatible.

```
# ./lib64/libc.so.6
GNU C Library stable release version 2.12, by Roland McGrath et al.
Copyright (C) 2010 Free Software Foundation, Inc.
…
 # cat /etc/redhat-release
Red Hat Enterprise Linux Server release 6.3 (Santiago)
```

This version is no longer current.

```
# /lib64/libc.so.6
GNU C Library stable release version 2.5, by Roland McGrath et al.
Copyright (C) 2006 Free Software Foundation, Inc.
…
# cat /etc/redhat-release
```

```
Red Hat Enterprise Linux Server release 5.6 (Tikanga)
```

If you see this error and cannot move to a more up to date platform you have the option of 2 simple workarounds that achieve the same aim:

1. Install HammerDB v2.13 in a directory parallel to your installation of HammerDB v2.14. Replace the hdb-components directory in the v2.13 installation with the hdb-components directory from v2.14. Remove the v.214 installation and run HammerDB from the v2.13 installation. Functionality will be the same as v2.14 but using TCL 8.5 instead of TCL 8.6.

2. Compile TCL and TK for your own platform as detailed below and merge the contents of the bin and lib directories under the HammerDB v21.4 installation directory with the contents of the bin and lib directories you have compiled whilst ensuring that you do not overwrite all of the contents in the lib directory (ie do not delete the directories first) . Alternatively you can also leave your compiled software in the default /usr/local directory. You may also need to recompile the extension for your chosen database.

Oracle on Windows Access blocked by UAC (User Account Control)

Microsoft UAC can prevent the running of applications. Additionally the security level may change after applying Microsoft updates and therefore an application that previously ran without issues may experience problems after an update has been applied.

In HammerDB if you observe an error reported in the Console on loading client libraries for your database, this may be due to 32/64-bit errors as detailed previously or on Windows can occur as security levels have been set too high. To prevent this happening you can either:

1. Disable UAC

2. Run HammerDB as Administrator.

To do this in the HammerDB bin directory right click the wish86t.exe executable.

Figure 19 wish85t.exe

Select Compatibility and choose Run this program as administrator.

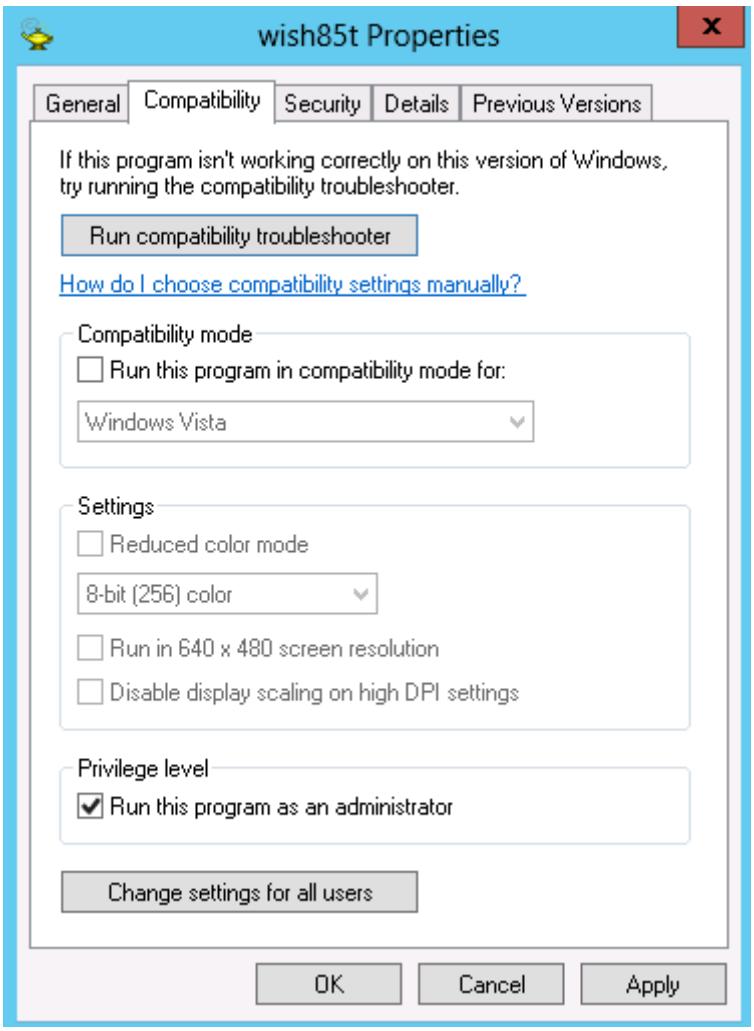

Figure 20 Run as Administrator

Click Apply and then OK.

Start HammerDB as normal by double clicking on the executable HammerDB.bat as described previously. HammerDB will start and a UAC prompt will ask:

"Do you want to allow the following program from an unknown publisher to make changes to this computer?" Program : wish85t.exe

Click Yes : HammerDB will start normally without the errors caused by UAC enabling access levels to load the Oracle libraries.

DEP prevents command line execution in Windows Server 2012

If you are running HammerDB commands at the command line such as with the autohammer extension, Data Execution Prevention (DEP) may prevent successful execution with and error such as "unhandled win32 exception occurred in tclsh85t.exe". In Windows Server 2012 DEP has now been enabled by default for all programs whereas on previous releases it was disabled by default. You can modify the DEP settings as shown to the default setting of previous Windows Server versions.

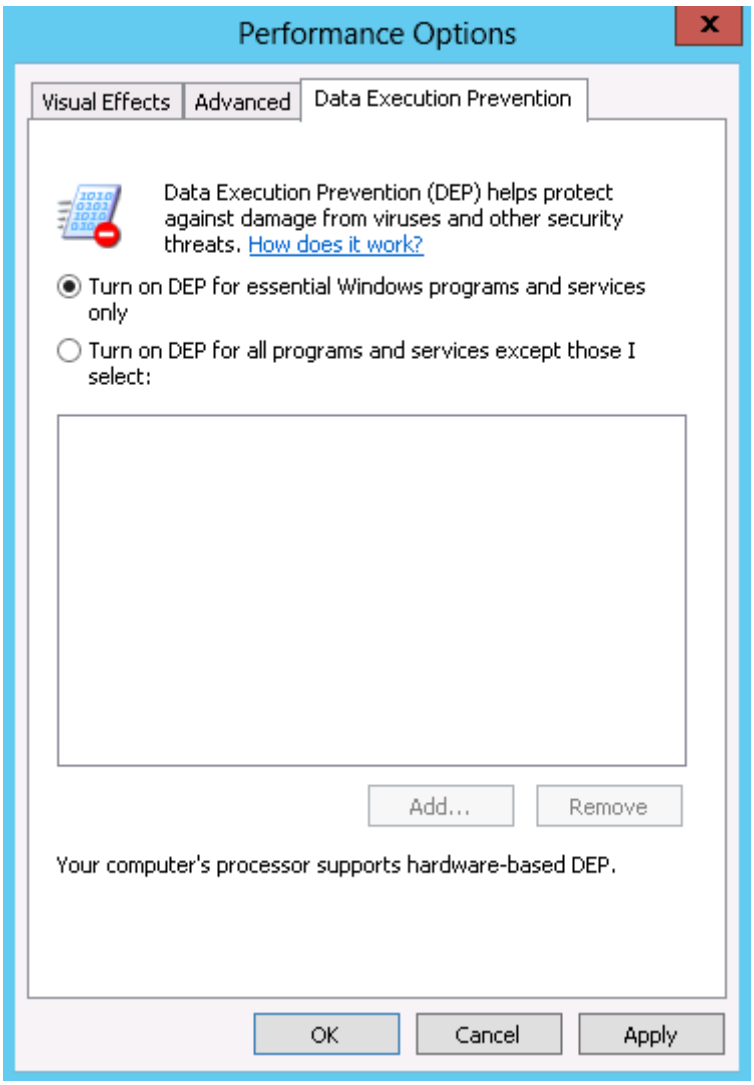

Figure 21 Disable DEP

Multithreaded OCI Applications Crash with Oracle 11*g* on Windows

If you are running HammerDB against Oracle 11g on Windows and the application crashes this is likely to be a known issues with the Oracle 11*g* Diagnosability Infrastructure that causes application crashes for multithreaded applications on Windows. Note that again this bug is an Oracle bug and not a HammerDB bug and can be investigated on the My Oracle Support website with the following reference.

Bug 12733000 OCIStmtRelease crashes or hangs if called after freeing the service context handle

To troubleshoot check your system logs for an application event showing a crash in orageneric11.dll such as the following:

```
Faulting application name: wish85t.exe, version: 8.5.2.9, time stamp: 0x4dda2c09
Faulting module name: orageneric11.dll, version: 11.2.0.2, time stamp:
0x4dc7c18d
Exception code: 0xc0000005
Fault offset: 0x00000000006ba349
```

Referring to the referenced crash dump files shows an error as follows.

Fatal Error in Wish

Unhandled exception: Code=c0000005 Flags=0
1EnumerateLoadedModules64 failed with error -1073741819

To resolve this Oracle issue add the following entry to your SQLNET.ORA file.

```
# This file is actually generated by netca. But if customers choose to
# install "Software Only", this file wont exist and without the native
# authentication, they will not be able to connect to the database on NT.

SQLNET.AUTHENTICATION_SERVICES = (NTS)
DIAG_ADR_ENABLED=OFF
DIAG_SIGHANDLER_ENABLED=FALSE
DIAG_DDE_ENABLED=FALSE
```

Building HammerDB on UNIX platforms

For Linux and Windows platforms using the precompiled installations of HammerDB is strongly recommended. If you wish to run HammerDB on a non Linux or Windows platform such as Solaris, HP-UX or AIX you can do so by compiling the TCL, TK, Tcl Threads (Note that for TCL 8.6 threads is included by default), Oratcl, MySQLTCL and PgTCL packages for your platform (all software is available on sourceforge). Support will not be given for self-compiled builds in UNIX environments or installations not using the installer packages on Windows and Linux systems. It is assumed that you would not wish to compile tclodbc for UNIX, ODBC connectivity to SQL Server strongly recommended through the pre-compiled Windows installation. Compile your downloaded software as follows:

TCL

```
cd tcl8.6/unix
./configure --enable-threads
make
make install
```

TK

Note: To compile TK on Linux the X development RPMS are required.

```
cd tk8.6/unix
./configure --enable-threads
make
make install
```

Oratcl

```
cd oratcl4.5
./configure --enable-threads
make
make install
```

MySQLTcl

```
cd mysqltcl-3.052
./configure --enable-threads -with-mysql-include=/u01/mysql/include -with-mysql-
lib=/u01/mysql/lib
```

```
make
make install
```

PgTcl

```
 cd pgtcl2.0.0
./configure -with-postgres-include=/pg/PostgresPlus/9.2AS/include -with-
postgres-lib=/pg/PostgresPlus/9.1AS/lib
make
make install
```

Redis

Copy the Redis directory from the lib directory from one of the installer packages.

By default your software will have been installed in the /usr/local/bin and /usr/local/lib directories. Take the installation for HammerDB on Linux or Windows and replace the bin and lib directories with the bin and lib directories you have just compiled. HammerDB will now run on your chosen platform.

HammerDB source code

By default all of the HammerDB source code is made available with each and every software release. Note that the source code is identical whether run on Windows or Linux, it is only the TCL engine that is platform specific. All of the source code is contained within the files in the hdb-components directory in human readable form made available under the GNU GENERAL PUBLIC LICENSE Version 2. The file hammerdb.tcl loads all of the components in the correct order. If wishing to compile TCL and the packages separately for your platform the files with the *.tcl extension are all the required code you need for HammerDB after you have compiled and installed the TCL software and extensions. Note that support cannot be provided for HammerDB on any other operating system apart from the pre-compiled Windows and Linux packages as test environments are not available. The platform independent HammerDB source code is also available in the GIT repository on the HammerDB sourceforge site from version 2.14 onwards and you can retrieve updates from here, however note that you will need to either use the contents of the bin and lib directories from a precompiled version or build your own executables and libraries for HammerDB to run.

Support and Questions

For help use the HammerDB Sourceforge forum available at the HammerDB sourceforge project.