Db2 11.1 for Linux, UNIX, and Windows

Db2 CI

Db2 11.1 for Linux, UNIX, and Windows

Db2 CI

Notice regarding this document

This document in PDF form is provided as a courtesy to customers who have requested documentation in this format. It is provided As-Is without warranty or maintenance commitment.

Contents

Figures

Tables

DB2CI application development

DB2CI is a callable SQL interface to the Db2® Version 9.7 database servers. It is a 'C' and 'C++' application programming interface for Db2 database access that uses function calls to connect to databases, manage cursors, and perform SQL statements.

Starting with Version 9.7 Fix Pack 1, you can use the DB2CI interface to access databases on Db2 Version 9.7 servers on any of the supported operating systems.

The DB2CI interface provides support for a number of Oracle Call Interface (OCI) APIs. This support reduces the complexity of enabling existing OCI applications so that they work with Db2 databases. The IBM® Data Server Driver for Db2CI is the driver for the DB2CI interface.

IBM Data Server Driver for Db2CI

The IBM Data Server Driver for Db2CI provides support for DB2CI application development.

The IBM Data Server Client includes the Db2CI driver. You need to install this client to install the Db2CI driver.

The Db2CI driver provides support for calls to the following OCI APIs.

Table 1. Db2CI driver support Part 1

Supported OCI APIs
obindps
obndrn
obreak
ocan
OCIArrayDescriptorAlloc
OCIArrayDescriptorFree
OCIAttrGet
OCIAttrSet
OCIBindArrayOfStruct
OCIBindByName
OCIBindByPos
OCIBindDynamic
OCIBreak
OCIClientVersion
OCIConnectionPoolCreate
OCIConnectionPoolDestroy
OCIDateAddDays
OCIDateAddMonths
OCIDateAssign
OCIDateCheck
OCIDateCompare
OCIDateDaysBetween
OCIDateFromText
OCIDateLastDay
OCIDateNextDay
OCIDateSysDate
OCIDateTimeAssign
OCIDateTimeCheck
OCIDateTimeCompare
OCIDateTimeConstruct
OCIDateTimeConvert
OCIDateTimeFromText
OCIDateTimeGetDate
OCIDateTimeGetTime
OCIDateTimeGetTimeZoneName
OCIDateTimeGetTimeZoneOffset
OCIDateTimeIntervalAdd
OCIDateTimeIntervalSub
OCIDateTimeSubtract
OCIDateTimeSysTimeStamp
OCIDateTimeToText
OCIDateToText
OCIDateZoneToZone
OCIDefineArrayOfStruct
OCIDefineByPos
OCIDefineDynamic
OCIDescribeAny
OCIDescriptorAlloc
OCIDescriptorFree
OCIDirPathAbort
OCIDirPathColArrayEntryGet
OCIDirPathColArrayEntrySet
OCIDirPathColArrayReset
OCIDirPathColArrayRowGet
OCIDirPathColArrayToStream
OCIDirPathDataSave
OCIDirPathFinish
OCIDirPathFlushRow
OCIDirPathLoadStream
OCIDirPathPrepare
OCIDirPathStreamReset
OCIEnvCreate
OCIEnvInit
OCIEnvNlsCreate
OCIErrorGet
OCIFileClose
OCIFileExists
OCIFileFlush
OCIFileGetLength
OCIFileInit
OCIFileOpen
OCIFileRead
OCIFileSeek
OCIFileTerm
OCIFileWrite
OCIHandleAlloc
OCIHandleFree
OCIInitialize
OCILdaToSvcCtx
OCILobAppend
OCILobAssign
OCILobClose
OCILobCopy
OCILobCopy2
OCILobCreateTemporary
OCILobDisableBuffering
OCILobEnableBuffering
OCILobErase
OCILobErase2
OCILobFlushBuffer
OCILobFreeTemporary
OCILobGetChunkSize
OCILobGetLength
OCILobGetLength2
OCILobGetStorageLimit
OCILobIsEqual
OCILobIsOpen
OCILobIsTemporary
OCILobLocatorAssign
OCILobLocatorIsInit
OCILobOpen
OCILobRead
OCILobTrim
OCILobTrim2
OCILobWrite
OCILogoff
OCILogon
OCILogon2
OCINlsEnvironmentVariableGet
OCINumberAbs
OCINumberAdd
OCINumberArcCos
OCINumberArcSin
OCINumberArcTan
OCINumberArcTan2
OCINumberAssign
OCINumberCeil
OCINumberCmp
OCINumberCos
OCINumberDec
OCINumberDiv
OCINumberExp
OCINumberFloor
OCINumberFromInt
OCINumberFromReal
OCINumberFromText
OCINumberHypCos
OCINumberHypSin
OCINumberHypTan
OCINumberInc
OCINumberIntPower
OCINumberIsInt
OCINumberIsZero
OCINumberLn
OCINumberLog
OCINumberMod
OCINumberMul
OCINumberNeg
OCINumberPower
OCINumberPrec
OCINumberRound
OCINumberSetPi
OCINumberSetZero
OCINumberShift
OCINumberSign
OCINumberSin
OCINumberSqrt
OCINumberSub
OCINumberTan
OCINumberToInt
OCINumberToReal
OCINumberToRealArray
OCINumberToText
OCINumberTrunc
OCIParamGet
OCIParamSet
OCIPasswordChange
OCIPing
OCIRawAllocSize
OCIRawAssignBytes
OCIRawAssignRaw
OCIRawPtr
OCIRawResize
OCIRawSize
OCIReset
OCIResultSetToStmt
OCIRowidToChar
OCIServerAttach

Table 2. Db2CI driver support Part 2

Supported OCI APIs		
OCIServerDetach		
OCIServerVersion	OCIThreadIdGet	odefinps
OCISessionBegin	OCIThreadIdInit	odescr
OCISessionEnd	OCIThreadIdNull	odessp
OCISessionGet	OCIThreadIdSame	oerhms
OCISessionPoolCreate	OCIThreadIdSet	oermsg
OCISessionPoolDestroy	OCIThreadIdSetNull	oexec
OCISessionRelease	OCIThreadInit	oexfet
OCIStmtExecute	OCIThreadIsMulti	oexn
OCIStmtFetch	OCIThreadJoin	ofen
OCIStmtFetch2	OCIThreadKeyDestroy	ofetch
OCIStmtGetBindInfo	OCIThreadKeyGet	oflng
OCIStmtGetPieceInfo	OCIThreadKeyInit	ogetpi
OCIStmtPrepare	OCIThreadKeySet	olog
OCIStmtPrepare2	OCIThreadMutexAcquire	ologof
OCIStmtRelease	OCIThreadMutexDestroy	onbclr
OCIStmtSetPieceInfo	OCIThreadMutexInit	onbset
OCIStringAllocSize	OCIThreadMutexRelease	onbtst
OCIStringAssign	OCIThreadProcessInit	oopen
OCIStringAssignText	OCIThreadTerm	oopt
OCIStringPtr	OCITransCommit	oparse
OCIStringResize	OCITransDetach	opinit
OCIStringSize	OCITransForget	orol
OCISvcCtxToLda	OCITransMultiPrepare	osetpi
OCITerminate	OCITransPrepare	SQLEnvGet
OCIThreadClose	OCITransRollback	sqlld2
OCIThreadCreate	OCITransStart	sqllda
OCIThreadHandleGet	oclose	SQLSvcCtxGet
OCIThreadHndDestroy	ocof	xaoEnv
OCIThreadHndInit	ocom	xaosterr
OCIThreadIdDestroy	ocon	xaoSvcCtx

Building DB2CI applications

You can build DB2CI applications using an existing Oracle Call Interface (OCI) application and the **`bldapp`** script file.

Before you begin

- You must have aDb2 database with the same structure as the Oracle database used by your existing OCI application.
- You must have installed the IBM Data Server Client.

About this task

Db2 samples provides a script called `bldapp` for compiling and linking applications that use OCI functions supported by the IBM Data Server Driver for Db2CI. It is located in the *`DB2DIR`*`\samples\db2ci` or *`DB2DIR`*`/samples/db2ci` directories, along with sample programs. *DB2DIR* represents the location where your Db2 copy is installed.

The `bldapp` script file takes up to four parameters. The first parameter, `$1`, specifies the name of your source file. The additional parameters are only required to build embedded SQL programs that requires a connection to the database: the second parameter, `$2`, specifies the name of the database to which you want to connect; the third parameter, `$3`, specifies the user ID for the database, and `$4` specifies the password. If the program contains embedded SQL, indicated by the .sqc extension, then the embprep script is called to precompile the program, producing a program file with a .c extension.

Restriction

- Ensure that your existing OCI application only has calls to OCI functions supported by the Db2CI driver. See "IBM Data Server Driver for Db2CI" on page 1 for a complete list of supported OCI functions.

Procedure

1. If you are building your DB2CI application using an existing OCI application, ensure that you specify the db2ci.h include file.
2. Build your DB2CI application with the bldapp script file The following example shows how to build the sample program tbinfo from the source file tbinfo.c on Linux and UNIX operating systems:

   ```
   cd $INSTHOME/sqllib/samples/db2ci
   bldapp tbinfo
   ```

   The result is an executable file, tbinfo.
3. Run the executable file generated in the previous step by entering the executable name as follows:

   ```
   tbinfo
   ```

DB2CI application compile and link options (AIX)

The compile and link options in this topic are recommended for building DB2CI applications with the AIX® IBM C compiler.

You can find the following options in the *DB2DIR*/samples/cli/bldapp batch file, where *DB2DIR* is the location where your Db2 copy is installed.

Compile options:

xlc The IBM C compiler.

$EXTRA_CFLAG
Contains the value "-q64" for 64-bit environments; otherwise, contains no value.

-I$DB2PATH/include
Specify the location of the Db2 include files. For example: $HOME/sqllib/include

-c Perform compile only; no link. This script has separate compile and link steps.

Link options:

xlc Use the compiler as a front end for the linker.

$EXTRA_CFLAG
Contains the value "-q64" for 64-bit environments; otherwise, contains no value.

-o $1 Specify the executable program.

$1.o Specify the object file.

utilci.o
Include the utility object file for error checking.

-L$DB2PATH/$LIB
: Specify the location of the Db2 runtime shared libraries. For example: `$HOME/sqllib/$LIB`. If you do not specify the `-L` option, the compiler assumes the following path: `/usr/lib:/lib`.

-ldb2ci
: Link with the DB2CI library.

DB2CI application compile and link options (Linux)

The compile and link options in this topic are recommended for building DB2CI applications with the GNU/Linux gcc compiler.

You can find the following options in the *DB2DIR*/samples/db2ci/bldapp batch file, where *DB2DIR* is the location where your Db2 copy is installed.

Compile options:

gcc The C compiler.

$EXTRA_C_FLAGS
: Contains one of the following:

- `-m31` on Linux for zSeries only, to build a 32-bit library;
- `-m32` on Linux for x86, x64 and POWER®, to build a 32-bit library;
- `-m64` on Linux for zSeries, POWER, x64, to build a 64-bit library; or
- No value on Linux for IA64, to build a 64-bit library.

-I$DB2PATH/include
: Specify the location of the Db2 include files. For example: `$HOME/sqllib/include`

-c Perform compile only; no link. Compile and link are separate steps.

Link options:

gcc Use the compiler as a front end for the linker.

$EXTRA_C_FLAGS
: Contains one of the following:

- `-m31` on Linux for zSeries only, to build a 32-bit library;
- `-m32` on Linux for x86, x64 and POWER, to build a 32-bit library;
- `-m64` on Linux for zSeries, POWER, x64, to build a 64-bit library; or
- No value on Linux for IA64, to build a 64-bit library.

-o $1 Specify the executable.

$1.o Include the program object file.

utilci.o
: Include the utility object file for error checking.

$EXTRA_LFLAG
: For 32-bit it contains the value "-Wl,-rpath,$DB2PATH/lib32", and for 64-bit it contains the value "-Wl,-rpath,$DB2PATH/lib64".

-L$DB2PATH/$LIB
Specify the location of the Db2 static and shared libraries at link-time. For example, for 32-bit: `$HOME/sqllib/lib32`, and for 64-bit: `$HOME/sqllib/lib64`.

-ldb2ci
Link with the DB2CI library.

DB2CI application compile and link options (Windows)

The compile and link options in this topic are recommended for building DB2CI applications with the Microsoft Visual C++ compiler.

You can find the following options in the *DB2DIR*`\samples\db2ci\bldapp.bat` batch file, where *DB2DIR* is the location where your Db2 copy is installed.

Compile options:

%BLDCOMP%
Variable for the compiler. The default is `cl`, the Microsoft Visual C++ compiler. It can be also set to `icl`, the Intel C++ Compiler for 32-bit and 64-bit applications, or `ecl`, the Intel C++ Compiler for Itanium 64-bit applications.

-Zi Enable debugging information.

-Od Disable optimizations. It is easier to use a debugger with optimization off.

-c Perform compile only; no link.

-W2 Set warning level.

-DWIN32
Compiler option necessary for Windows operating systems.

Link options:

link Use the linker.

-debug Include debugging information.

-out:%1.exe
Specify the executable.

%1.obj Include the object file.

db2ci.lib or db2ci64.lib
Link to the DB2CI library. For Windows 32-bit operating systems, use `db2ci.lib`. For Windows 64-bit operating systems, use `db2ci64.lib`.

Refer to your compiler documentation for additional compiler options.

Index

Printed in USA