DB2® Everyplace(R)
Enterprise Edition

Version 9.1

Application Developer's Quick Start Guide

This information is intended to get you started with basic Db2(R) Everyplace(R) database applications and base applications to perform synchronization for DB2 Everyplace Enterprise Edition. After you are familiar with the basic DB2 Everyplace applications, you can customize them to meet your needs. This guide includes all of the steps that are needed to execute and debug a simple application (either the sample applications or your own basic application).

The C/C++ programming language is used in all of the examples in this information.

Product Overview

IBM(R) DB2 Everyplace is a relational database and synchronization server that extends enterprise applications and data to mobile devices such as personal digital assistants (PDAs) and smart phones. DB2 Everyplace Enterprise Edition provides a complete data-centric mobile synchronization system. The secure synchronization server manages distribution and synchronization of data to mobile workers. DB2 Everyplace Enterprise Edition integrates with many backend data sources. To develop a DB2 Everyplace application, you need to use the appropriate development tools for a certain operating system, and include appropriate header files and libraries. Then, write a database application program by using DB2 Everyplace CLI. To write synchronization programs, use the DB2 Everyplace Sync Client Interface. You can look at the sample code and get an idea of how to build your own application.

1 Integrated development environment (IDE) for the corresponding operating systems

Depending on the operating system of the mobile device, you will need a particular integrated development environment (IDE).

For the detailed list of supported tools and instructions, see the following information: C/C++ development tools.

Palm OS

Metrowerks CodeWarrior Development Studio for Palm OS

Symbian OS

- CodeWarrior for Symbian OS
- Microsoft(R) Visual C++ 6

Windows(R) CE

PocketPC 2000

Microsoft eMbedded Visual Tools 3.0, Pocket PC SDK
Emulator: X86EM processor
Device: ARM, MIPS, or SH3 processor

PocketPC 2002

Microsoft eMbedded Visual Tools 3.0, Pocket PC 2002 SDK
Emulator: X86 processor
Device: ARM processor

PocketPC 2003

Microsoft eMbedded Visual Tools 4.0, Pocket PC 2003 SDK
Emulator: X86 processor
Device: ARMV4 processor

Windows Mobile 2005

Microsoft eMbedded Visual Tools 4.0 or Visual Studio 2005
Emulator: ARM-based processor
Device: ARMV4, or ARMV4T where T stands for Thumb

The following platforms support the Microsoft .NET Compact Framework:

- PocketPC 2002 (Visual Studio .NET)
- PocketPC 2003 (Visual Studio .NET)
- Windows Mobile 2005 (Visual Studio 2005)

Restriction:

The DB2 Everyplace Sync Client libraries do not support Microsoft eMbedded Visual Basic.

Linux[R]

- You can use your embedded Linux distribution's cross platform development tools to develop your applications. The embedded Linux kernel must support ELF binaries enabled.
- If you are developing the application on a system that has the same architecture as the target system, you can also use the GNU C/C++ tools.

2 Setting up a project and workspace for an application

This procedure varies depending on the development tools and operating system that you are developing for. Refer to the documentation that comes with the development tools you are using for instructions.

3 Developing DB2 Everyplace database applications

This topic presents how to develop a simple database application and the troubleshooting steps.

Required header and library files

After setting up a project, you need certain header files and library files in order for the application to run successfully.
Note: In the following file paths, *lang* is the four letter code of your language and <DSYPATH> is where DB2 Everyplace is installed.
Include the following DB2 Everyplace header files in the project.

<DSYPATH>\Clients\include\sqlcli.h
<DSYPATH>\Clients\include\sqlcli1.h
<DSYPATH>\Clients\include\sqlext.h
<DSYPATH>\Clients\include\sqlsystm.h

The following list summarizes the DB2 Everyplace libraries and lists additional information for each operating system. For more detailed information, see the DB2 Everyplace version 9.1 InfoCenter.

- **Palm OS:** <DSYPATH>\clients\palmos\database\DB2e.lib
- **Symbian OS:**
  Emulator applications: <DSYPATH>\clients\Symbian7\database\wins\DB2e.lib
  Device applications: <DSYPATH>\clients\Symbian7\database\armi\DB2e.lib
- **Windows CE:**
  ARM processor:
  V3.00: <DSYPATH>\clients\wince\database\v3\armrel\DB2e.lib
  V4.00: <DSYPATH>\clients\wince\database\v4\ARM4VRel\DB2e.lib
  MIPS processor:
  V3.00: <DSYPATH>\clients\wince\database\v3\mipsrel\DB2e.lib
  V4.00: <DSYPATH>\clients\wince\database\v4\MIPSIVRel\DB2e.lib
  Windows CE emulator:
  V3.00: <DSYPATH>\clients\wince\database\v3\x86emrel\DB2e.lib (for Pocket PC emulator)
  <DSYPATH>\clients\wince\database\v4\x86rel\DB2e.lib (for Pocket PC 2002 emulator)
  V4.00: <DSYPATH>\clients\wince\database\v4\emulatorRel\DB2e.lib (for WinCE.NET emulator)
  Verify that UNICODE is enabled for the project. Add UNICODE and _UNICODE to the **Preprocessor Definition** of the Project Settings.
  XScale processor: V4.00: <DSYPATH>\clients\wince\database\v4\XScaleRel\DB2e.lib
- **Linux:** $DSYINSTDIR/Clients/Linux/database/proc/libdb2e.so

Required DB2 Everyplace files for testing

Copy the DB2 Everyplace libraries to the mobile device or the emulator for your operating system. Without these files, a DB2 Everyplace application will not load. For more information, see Testing a C/C++ application.

Sample applications

The samples illustrate how to build a basic DB2 Everyplace application. You can customize them to meet your needs.
Note: By using the sample code examples, you agree to the terms of the "Code license and disclaimer information".
Download the sample code at: DB2 Everyplace sample code.

Related information
Sample application code in the Information Center
The sample DB2 Everyplace Sync Client C/C++ application

Checking whether the SQL is successful

Check the return value from the API SQLRETURN.

- If the value is SQL_SUCCESS or SQL_SUCCESS_WITH_INFO, the SQL statement has been executed successfully.
- If the value is other than SQL_SUCCESS or SQL_SUCCESS_WITH_INFO, for example SQL_ERROR and SQL_INVALID_HANDLE, the execution of the SQL statement has failed.

Troubleshooting your application

If the execution of SQL statements fails, follow these steps to troubleshoot.

- Make sure that you include all the libraries that are needed for a DB2 Everyplace application.

- For messages that indicate errors in functions, first determine if the errors are caused by errors in handles and function parameters. These two types of errors are the most common ones.
- Use the SQLGetDiagRec() function to get the current value of the SQLSTATE field of a diagnostic record that contains error, warning, and status information. For the information on this function's syntax and usage, refer to SQLGetDiagRec--Get multiple fields settings of diagnostic record.
- For the SQLState messages that are returned by SQL, refer to SQLState messages reported by SQL.

4 Developing DB2 Everyplace Sync Client applications

This topic presents how to develop a simple synchronization application and the troubleshooting steps.

Required header and library files

After setting up a project, you need certain header files and library files in order for the application to run successfully.
Note: In the following file paths, *lang* is the four letter code of your language and <DSYPATH> is where DB2 Everyplace is installed.
Include the following DB2 Everyplace header files in the project.
<DSYPATH>\Clients\include\isyncore.h
The following list summarizes the DB2 Everyplace libraries and lists additional information for each operating system. For more detailed information, see the DB2 Everyplace version 9.1 InfoCenter.

- **Palm OS:**
  - To compile the application, include the following files in the project:
    <DSYPATH>\Clients\palmos\sync\isyncore.lib
    <DSYPATH>\Clients\palmos\sync\isyncconf.lib
  - When running the application, copy the following files to your target device.
    All the .prc files in <DSYPATH>\Clients\palmos\sync\
    All the .prc files in <DSYPATH>\Clients\palmos\database\
- **Symbian OS:**
  - To compile the application, include the following files in the project:
    <DSYPATH>\ Clients\symbian7\sync\lang\v3\armi | wins\isyncore.lib
    <DSYPATH>\ Clients\symbian7\sync\lang\v3\armi | wins\isyncconf.lib
  - When running the application, copy the following files to your target device.
    All the .dll and .jar files in <DSYPATH>\Clients\sybmian7\sync\lang\armi|wins\
    All the .dll files in <DSYPATH>\Clients\symbian7\database\armi|wins\
- **Windows CE:**
  To compile the application, include the following files in the project:
  - Pocket PC 2000 and Pocket PC 2002:
    <DSYPATH>\ Clients\wince\sync\lang\v3\ARMRel | MIPSRel | X86EMRel |X86Rel\isyncore.lib
    <DSYPATH>\ Clients\wince\sync\lang\v3\ARMRel | MIPSRel | X86EMRel |X86Rel\isyncconf.lib
  - Pocket PC 2003 and Windows Mobile 2005:
    <DSYPATH>\Clients\wince\sync\lang\v4\ARMV4Rel | ARMV4TRel | emulatorRel | MIPSIVRel | X86Rel | XScaleRel \isyncore.lib
    <DSYPATH>\Clients\wince\sync\lang\v4\ARMV4Rel | ARMV4TRel | emulatorRel | MIPSIVRel | X86Rel | XScaleRel \isyncconf.lib

  When running the application, copy the following files to your target device.
  - Pocket PC 2000 and Pocket PC 2002:
    - All the .dll and .jar files in <DSYPATH>\Clients\wince\sync\lang\v3\ARMRel | MIPSRel | X86EMRel | X86Rel\
    - All the .dll files in <DSYPATH>\Clients\wince\database\v3\ARMDev | ARMRel | MIPSDev | MIPSRel | X86Dev | X86EMDev | X86EMRel | X86Rel\
  - Pocket PC 2003:
    - All the .dll and .jar files in <DSYPATH>\ Clients\wince\sync\lang\ v4\ARMV4Rel | ARMV4TRel | emulatorRel| MIPSIVRel | X86Rel | XScaleRel\
    - All the .dll files from <DSYPATH>\Clients\wince\database\v4\ARMV4Dev | ARMV4Rel | ARMV4TDev | ARMV4TRel | emulatorDev | emulatorRel | MIPSIVDev | MIPSIVRel | SH4Dev | SH4Rel | X86Dev | X86Rel | XScaleDev | XScaleRel\

Sample applications

The samples illustrate how to build a basic DB2 Everyplace application. You can customize them to meet your needs.
Note: By using the sample code examples, you agree to the terms of the "Code license and disclaimer information. "
Download the sample code at: DB2 Everyplace sample code.

Related information
Sample application code in the Information Center
The sample DB2 Everyplace Sync Client C/C++ application

Checking whether the synchronization is successful

Check the return value from the Sync Client API isy_INT32 iscEngineSync(HISCENG hSyncer).

- If the value is ISCRTN_Succeeded (=0), the synchronization is successful.
- If the value is ISCRTN_Failed (=-1), the synchronization fails.

Troubleshooting your application

If the synchronization fails, use the resources in this topic to troubleshoot.

- See the trace file trace-isyn in your target directory.
- Refer to DB2 Everyplace support and troubleshooting.

5 Code license and disclaimer information

? IBM grants you a nonexclusive copyright license to use all programming code examples from which you can generate similar function tailored to your own specific needs.

SUBJECT TO ANY STATUTORY WARRANTIES WHICH CANNOT BE EXCLUDED, IBM, ITS PROGRAM DEVELOPERS AND SUPPLIERS MAKE NO WARRANTIES OR CONDITIONS EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT, REGARDING THE PROGRAM OR TECHNICAL SUPPORT, IF ANY.

UNDER NO CIRCUMSTANCES IS IBM, ITS PROGRAM DEVELOPERS OR SUPPLIERS LIABLE FOR ANY OF THE FOLLOWING, EVEN IF INFORMED OF THEIR POSSIBILITY:

1. LOSS OF, OR DAMAGE TO, DATA;
2. DIRECT, SPECIAL, INCIDENTAL, OR INDIRECT DAMAGES, OR FOR ANY ECONOMIC CONSEQUENTIAL DAMAGES; OR
3. LOST PROFITS, BUSINESS, REVENUE, GOODWILL, OR ANTICIPATED SAVINGS.

SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OR LIMITATION OF DIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, SO SOME OR ALL OF THE ABOVE LIMITATIONS OR EXCLUSIONS MAY NOT APPLY TO YOU.