IBM solidDB
Version 7.0

Getting Started Guide

SC27-3838-05

Note

Before using this information and the product it supports, read the information in “Notices” on page 77.

First edition, fifth revision

This edition applies to V7.0 Fix Pack 8 of IBM solidDB (product number 5724-V17) and to all subsequent releases and modifications until otherwise indicated in new editions.

Contents

Figures

Tables

Summary of changes

Changes for revision 05

- Section System requirements updated: support for Solaris 11 introduced in Fix Pack 6.

Changes for revision 04

- Editorial corrections.

Changes for revision 03

- Section Installing solidDB® ODBC Driver updated: you can use the installation program to install the ODBC driver only on Windows environments. On Linux and UNIX environments, to install the ODBC driver to a different machine than where solidDB is installed, you must copy the ODBC driver library files manually.
- Section System requirements updated: support for Windows 8 and Windows Server 2012 introduced in Fix Pack 4.

Changes for revision 02

- Editorial corrections.

Changes for revision 01

- Section System requirements updated: support for System z® Linux introduced in Fix Pack 1.

About this manual

This guide provides an overview of the IBM® solidDB product family. It also includes instructions for how to set up solidDB for the first time and how to run samples.

This guide does not provide instructions for how to install and configure solidDB Universal Cache or how to start using the different features and functionality available with the solidDB products. For information about how the documentation is structured, see section Documentation in this guide.

Typographic conventions

solidDB documentation uses the following typographic conventions:

Table 1. Typographic conventions

Format	**Used for**
Database table	This font is used for all ordinary text.
NOT NULL	Uppercase letters on this font indicate SQL keywords and macro names.
`solid.ini`	These fonts indicate file names and path expressions.

`SET SYNC MASTER YES;`
`COMMIT WORK;`
	This font is used for program code and program output. Example SQL statements also use this font.

`run.sh`	This font is used for sample command lines.
TRIG_COUNT()	This font is used for function names.
java.sql.Connection	This font is used for interface names.
`LockHashSize`	This font is used for parameter names, function arguments, and Windows registry entries.
argument	Words emphasized like this indicate information that the user or the application must provide.
Administrator Guide	This style is used for references to other documents, or chapters in the same document. New terms and emphasized issues are also written like this.
File path presentation	Unless otherwise indicated, file paths are presented in the UNIX format. The slash (/) character represents the installation root directory.

Table 1. Typographic conventions (continued)

Format	**Used for**
Operating systems	If documentation contains differences between operating systems, the UNIX format is mentioned first. The Microsoft Windows format is mentioned in parentheses after the UNIX format. Other operating systems are separately mentioned. There may also be different chapters for different operating systems.

Syntax notation conventions

solidDB documentation uses the following syntax notation conventions:

Table 2. Syntax notation conventions

Format	**Used for**
`INSERT INTO` *`table_name`*	Syntax descriptions are on this font. Replaceable sections are on *this* font.
`solid.ini`	This font indicates file names and path expressions.
[ ]	Square brackets indicate optional items; if in bold text, brackets must be included in the syntax.
|	A vertical bar separates two mutually exclusive choices in a syntax line.
{ }	Curly brackets delimit a set of mutually exclusive choices in a syntax line; if in bold text, braces must be included in the syntax.
...	An ellipsis indicates that arguments can be repeated several times.
. . .	A column of three dots indicates continuation of previous lines of code.

Accessibility

Accessibility features help users with physical disabilities, such as restricted mobility or limited vision, to use software products successfully. The following sections specify the major accessibility features in solidDB.

Keyboard input and navigation

Keyboard input

You can operate solidDB using only the keyboard. You can use keys or key combinations to perform all operations. Standard operating system keystrokes are used for standard operating system operations. Standard operating system keyboard accessibility features are supported by solidDB. For more information about using keys or key combinations to perform operations, see the appropriate operating system documentation.

Keyboard navigation

You can navigate the solidDB user interface using keys or key combinations only. For more information about using keys or key combinations to navigate the command line interfaces, see the appropriate operating system documentation.

Accessible display

solidDB supports standard operating system display settings, such as high contrast and font settings.

Font settings

You can select the color, size, and font for the text using standard operating system operations. For more information about specifying font settings, see the appropriate operating system documentation.

Non-dependence on color

You do not need to distinguish between colors in order to use any of the functions in solidDB.

Compatibility with assistive technologies

solidDB interacts with the operating system through standard APIs which support interaction with assistive technologies, which enables you to use screen readers and other accessibility tools.

Accessible documentation

Documentation for solidDB is provided in HTML format via the Information Center, which is viewable in most Web browsers. Information Center allows you to view documentation according to the display preferences set in your browser. It also allows you to use screen readers and other assistive technologies.

See the solidDB Information Center opening page for more details on the Information Center accessibility.

Part 1. IBM solidDB product overview

IBM solidDB product family features relational, in-memory database technology that delivers extreme speed, performing up to ten times faster than conventional, disk-based databases. IBM solidDB, or simply, solidDB, uses the familiar SQL language to allow applications to achieve tens of thousands of transactions per second with response times measured in microseconds.

IBM solidDB 7.0 database software includes two capabilities:

IBM solidDB is a fully featured, relational, in-memory database that delivers extreme speed and extreme availability to meet performance and reliability demands of real-time applications. It provides the ability to have both in-memory tables and on-disk tables within a single solidDB instance. Other features include a High Availability implementation and several different replication topologies.

IBM solidDB Universal Cache is a relational, in-memory caching software that accelerates traditional disk-based relational database servers by caching performance critical data into one or more solidDB in-memory database instances.

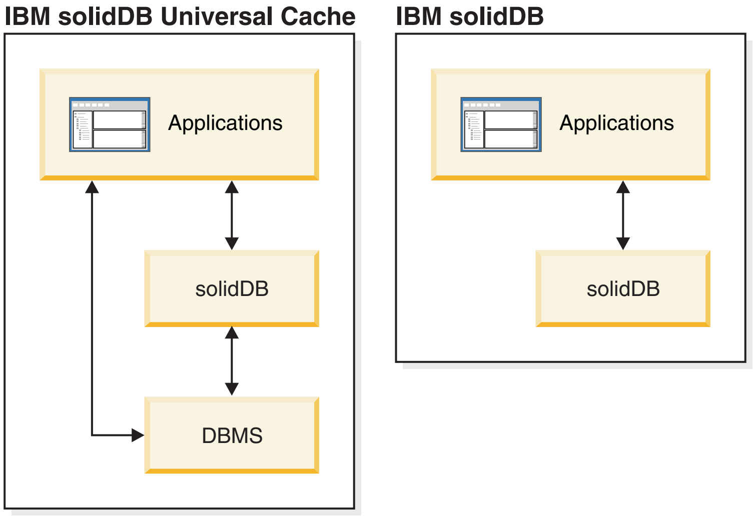


Figure 1. IBM solidDB 7.0

1 IBM solidDB

IBM solidDB is a relational database server that combines the high performance of in-memory tables with the nearly unlimited capacity of disk-based tables.

Pure in-memory databases are fast, but strictly limited by the size of memory. Pure disk-based databases allow nearly unlimited amounts of storage, but their performance is dominated by disk access. Even if the computer has enough memory to store the entire database in memory buffers, database servers that are designed for disk-based tables can be slow because the data structures that are optimal for disk-based tables are far from being optimal for in-memory tables. The solidDB solution is to provide a single hybrid database server that contains two optimized engines inside it:

- The main-memory engine (MME) is optimized for in-memory access.
- The disk-based engine (DBE) is optimized for disk-based access.

Both engines coexist inside the same server process, and a single SQL statement can access data from both engines.

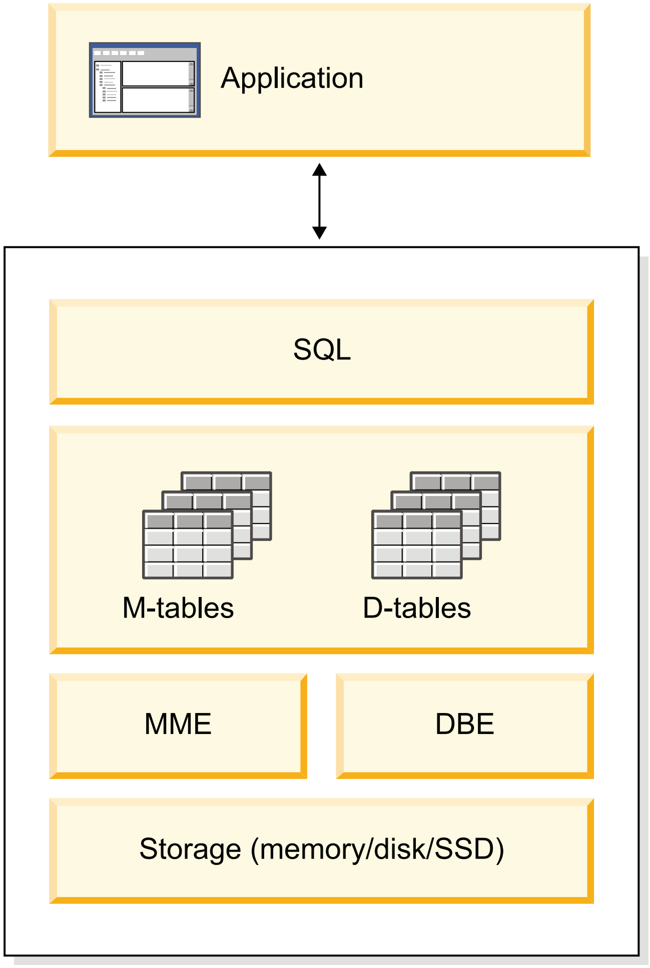


Figure 2. IBM solidDB hybrid server architecture

In addition to a fully functional relational database server, solidDB provides synchronization features that allow updated data in one solidDB to be sent to one or more other solidDB instances.

The solidDB server can also be configured for high availability. The HotStandby component enables a secondary server (a standby server) to run in parallel with the primary server (an active server) and keep an up-to-date copy of the data in the primary server.

You can also link your client application directly to the database server routines for higher performance and tighter control over the server. The direct linking capabilities are called shared memory access (SMA) or linked library access (LLA).

1.1 Architectural overview of solidDB

The solidDB database server uses a client/server model; the solidDB deployment consists of cooperating server and client processes. The server process manages the database files, accepts connections to the database from client applications, and carries out actions on the database as requested by the clients.

The client process is used to pass the required tasks (through the server process) to the database. There can be several client types: a client can be a command-line tool, a graphical application, or a database maintenance tool. Typically, different applications act as clients to connect to solidDB.

The client and the server can be on different hosts (nodes), in which case they communicate over a network. solidDB provides simultaneous support for multiple network protocols and connection types. Both the database server and the client applications can be simultaneously connected to multiple sites using multiple different network protocols.

solidDB can also run within the application process. This is provided by solidDB *shared memory access* (SMA) and *linked library access* (LLA). In this case, the application is linked to a function library that is provided with the product. The linked application communicates with the server by using direct function calls, thus skipping the overhead that is required when the client and server communicate through network protocols such as the TCP/IP. By replacing the network connection with local function calls, performance is improved significantly.

To submit a query (an SQL statement) to a database server, a client must be able to communicate with that database server. solidDB, like many other database servers, uses *drivers* to enable this communication. Client applications call functions in the driver, and the driver then handles the communications and other details with the server. For example, you might write a C program that calls functions in the (ODBC) driver, or you might write a Java™ program that calls functions in the (JDBC) driver.

1.2 Main features and functionality

The key features and functionality of the solidDB database product include the hybrid, relational database server with in-memory and disk-based tables, direct linking of applications to bypass performance-limiting network protocols, and various replication capabilities including high availability.

In-memory tables

In-memory tables store all their data in main memory in data-structures which are optimized for main-memory access. The benefit of the in-memory residency is low query latency and high throughput. Depending on the application needs, the in-memory data tables can be configured to be persistent or transient. In addition, flexible logging capabilities let you define the level of transactional durability that is required for persistent tables, including full durability.

For more information, see the *IBM solidDB In-Memory Database User Guide*.

Disk-based tables

Disk-based tables are traditional relational database tables, equivalent to those found in other disk-based database products. In addition to in-memory tables, normal disk-based tables can be used in the database too. Typically only a part of the data from disk-based tables is resident in main-memory at any point in time. Queries can span in-memory and disk-based tables. For example, an SQL join operation can join an in-memory table with a disk-based table; the actual location of the table is not apparent to the user.

The disk-based tables, combined with the small footprint and unattended operation capability, make the solidDB server ideal for embedded systems. Despite the small size, the product supports many advanced features like triggers, events, and stored procedures.

Direct linking with shared memory access (SMA) and linked library access (LLA)

Shared memory access (SMA) and linked library access (LLA) allow applications to link to solidDB server directly, without the need to communicate through network protocols such as TCP/IP. With SMA, you can link multiple applications while with LLA, you can link one application. By replacing the network connection with local function calls, performance is improved significantly.

For more information, see the *IBM solidDB Shared Memory Access and Linked Library Access User Guide*.

Replication technologies

In solidDB product family, data replication can be implemented with three different technologies, namely, Advanced Replication, InfoSphere® CDC Replication, and solidDB High Availability (HotStandby).

Replication can be synchronous or asynchronous. In synchronous replication, the data is durably delivered to the receiving end within the transaction that is executed at the originating end. In asynchronous replication, the data is delivered to the receiving end after the original transaction is committed. Replication can be based on a push or pull operational model. In the push model, it is the originating end that activates the replication. In the pull model, the receiving end takes the initiative.

solidDB High Availability (HotStandby)

solidDB High Availability, or simply HA or HotStandby, increases the availability of data. Servers are paired up in such a way that, if one fails, the other can take over. Data from the primary server is replicated to the secondary server, which is used as a 'hot standby' unit. If the primary server is unavailable, for example, because of hardware failure or scheduled maintenance, applications can connect to the secondary server and continue immediately without any loss of committed transactions.

In solidDB HA, all the data changes in the primary are propagated to the secondary using a push-based replication protocol. The protocol can be set to synchronous (*2-Safe*) or asynchronous (*1-Safe*). In principle, the transaction load is served at the primary. Should the primary fail, the secondary takes up the job (executes a failover) and continues to serve the load as the new primary. With the synchronous replication protocol, there is no risk of data loss during the failover.

Transparent Connectivity, a special connectivity mode of solidDB ODBC and JDBC drivers, offers the applications failover transparency and transparent load balancing between the primary and secondary.

Both relaxed and strict durability can be used with HotStandby. There is also a durability level called *adaptive*, which uses relaxed durability when both primary and secondary servers are active, and which switches to strict durability when the primary runs alone. With adaptive durability, the transactions are secured against any single failure, at any time.

In comparison to advanced replication and InfoSphere CDC replication, with solidDB HA, all the data in the database is always replicated. There are a few dynamic controls available, in the form of specialized SQL commands.

For more information, see the *IBM solidDB High Availability User Guide*.

Advanced replication between solidDB servers

Advanced replication technology represents an asynchronous pull-based approach. It allows users to distribute and synchronize, occasionally, data across multiple database servers. Advanced replication uses a master-replica model in which a single node holds the master copy of the data. One or more replica nodes can also have a copy of part or all of the master data. Furthermore, each replica can also have data that is specific to that replica only.

Replication is bidirectional; replica nodes can upload data to the master as well as download from it. If replicas submit conflicting data, the master can reject the data or alter the data before it makes the data available to all replicas. The management of replicated data is based on a flexible publish-subscribe model. The management interface is in the form of proprietary extensions to the SQL language.

The asynchronous approach in advanced replication setups means that the system is inherently flexible. If some nodes are down for any reason, those nodes can resynchronize when they come back up or reconnect to the system. This allows mobile devices, such as PDAs or notebooks, to request updated data, disconnect from the network, and then reconnect later. Users can choose how frequently they want to synchronize data.

For more information, see the *IBM solidDB Advanced Replication User Guide*.

InfoSphere CDC technology and universal caching

IBM InfoSphere Change Data Capture (InfoSphere CDC) technology can be used to implement asynchronous replication among various databases. InfoSphere CDC technology is based on an asynchronous push model. Unidirectional subscriptions can be created for real-time propagation of data changes from the source side to the target side. Bidirectional capability is achieved by setting up two subscriptions with mirrored source and target definitions.

- **With solidDB**, the InfoSphere CDC technology can be used for replicating data between any pair of solidDB server instances. For example, it enables geographic redundancy, or, it allows two solidDB instances to be configured in an active-active setup whereby two copies of the same data can be equally processed at either side.

  For more information, see the *IBM solidDB Replication with InfoSphere CDC User Guide*.
- **With solidDB Universal Cache**, the InfoSphere CDC technology can be used to set up a solidDB frontend cache that accelerates access to performance critical data stored in backend data servers.

  For more information, see the *IBM solidDB Universal Cache User Guide*.

In both setups, each solidDB server instance can also be a solidDB HotStandby pair.

InfoSphere CDC technology includes database-specific components that run as separate processes called InfoSphere CDC instances. A Management Console for managing the replication subscriptions and data mapping is also included.

1.3 solidDB data management components

This section describes the underlying components and processes that make solidDB the solution for managing distributed data in complex distributed system environments. It provides background information necessary to administer and maintain solidDB in your network environment.

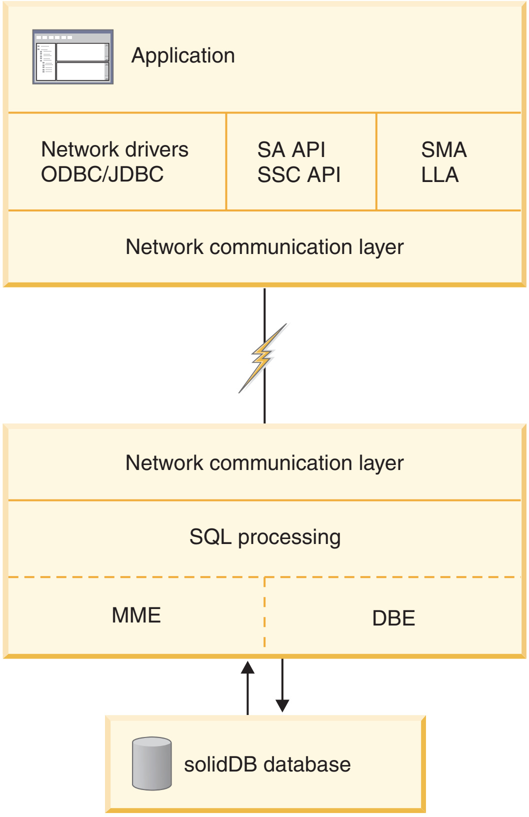


Figure 3. solidDB components

1.3.1 Database access methods and network drivers

Applications can connect to the solidDB server by using network drivers or by linking to the server directly.

In network-based access methods, the applications and the solidDB server are separate programs, typically communicating through the solidDB ODBC Driver or solidDB JDBC Driver.

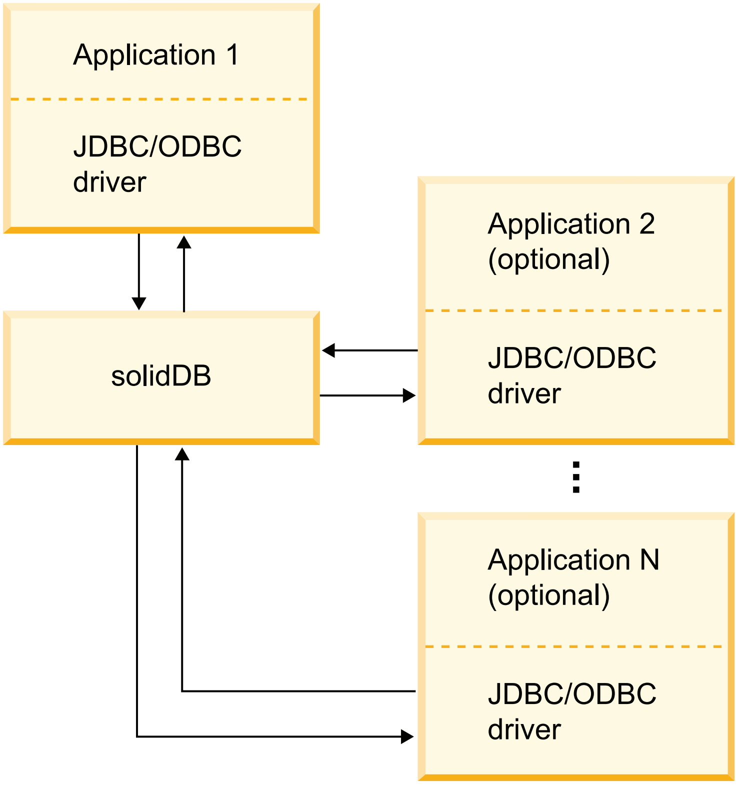


Figure 4. Client/server access method

Direct linking is provided through linked library access (LLA) and shared memory access (SMA). SMA and LLA are implemented as library files that contain a complete copy of the solidDB server in a library form. The SMA and LLA servers can also handle requests from remote applications which connect to the server through network communication protocols such as TCP/IP. The remote applications see the SMA or LLA server as similar to any other solidDB server; the local SMA and LLA applications see a faster, more precisely controllable version of the solidDB server.

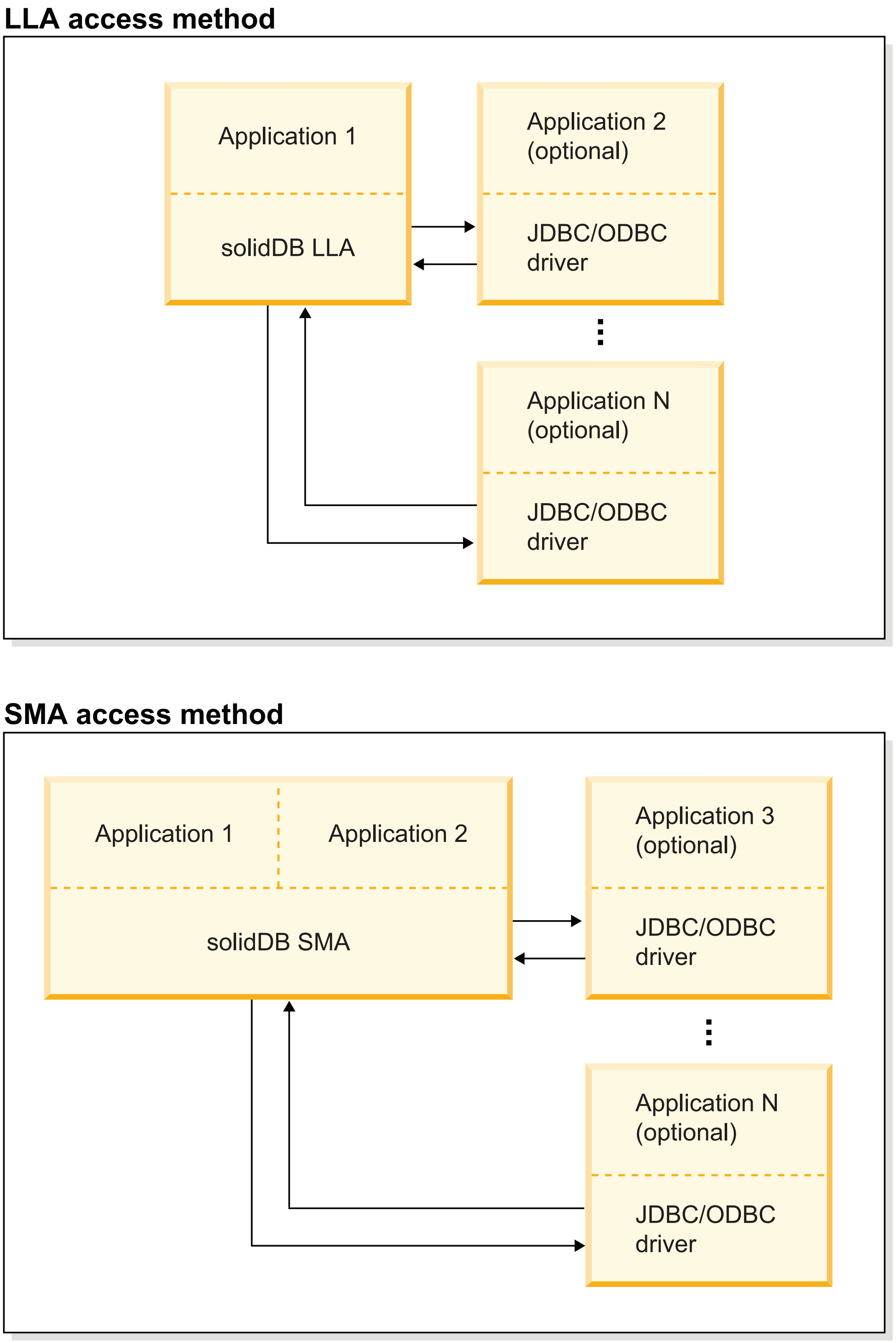


Figure 5. SMA and LLA access methods

ODBC

The solidDB ODBC Driver conforms to the Microsoft ODBC 3.51 API standard. solidDB ODBC Driver supported functions are accessed with solidDB ODBC API, a Call Level Interface (CLI) for solidDB databases, which is compliant with ANSI X3H2 SQL CLI.

For more details on the solidDB ODBC Driver, see *IBM solidDB Programmer Guide*.

JDBC

The solidDB JDBC Driver allows Java applications to access the database by using JDBC. The solidDB JDBC Driver implements most of the JDBC 2.0 specification.

For more details on the solidDB JDBC Driver, see *IBM solidDB Programmer Guide*.

Proprietary interfaces

solidDB also provides two proprietary interfaces, solidDB Application Programming Interface (SA API) and solidDB Server Control API (SSC API). These interfaces allow, for example, C programs to directly call functions inside the database server. These proprietary interfaces are provided with the linked library access (LLA) and solidDB shared memory access (SMA) libraries.

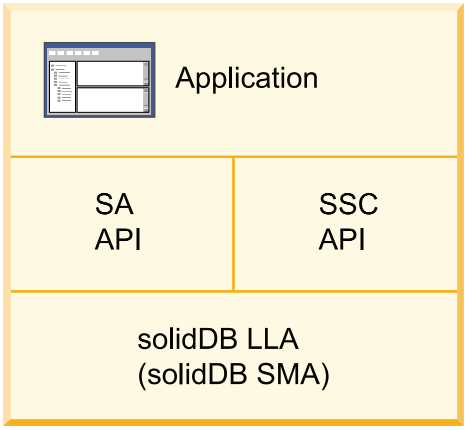


Figure 6. solidDB proprietary programming interfaces

Note: The SMA library supports only a subset of the SA API and SSC API functions.

solidDB Application Programming Interface (SA API)

solidDB SA is a low-level C-language client library for accessing data in solidDB database tables. The library contains about 90 functions that provide low-level mechanisms for connecting the database and running cursor-based operations.

In most cases, you can use the standards-based interfaces, such as ODBC or JDBC. However, in environments with heavy write loads (BATCH INSERTS AND UPDATES), solidDB SA can provide a significant performance advantage.

For more information about solidDB SA, see the *IBM solidDB Programmer Guide*.

solidDB Server Control API (SSC API)

The solidDB Server Control API (SSC API) is a set of functions that provide simple and efficient means to control the tasking system of a solidDB. For example, the SSC API functions are used to start and stop the linked library access (LLA) and shared memory access (SMA) server.

For more information about SSC API, see the IBM solidDB Shared Memory Access and Linked Library Access User Guide.

solidDB Server Control API (SSC API) for Java

The solidDB Server Control API (SSC API) for Java is a proprietary API, named after SolidServerControl class. The SSC API for Java calls are used to start and stop the linked library access (LLA) server. The actual database connections are done with normal solidDB JDBC API. Both the SSC API for Java classes and solidDB JDBC driver classes are included in the solidDB JDBC driver (`SolidDriver2.0.jar`).

For more information about SSC API for Java, see the IBM solidDB Shared Memory Access and Linked Library Access User Guide.

1.3.2 solidDB server components

The solidDB server processes the data requests submitted through solidDB SQL. The server stores data and retrieves it from the database.

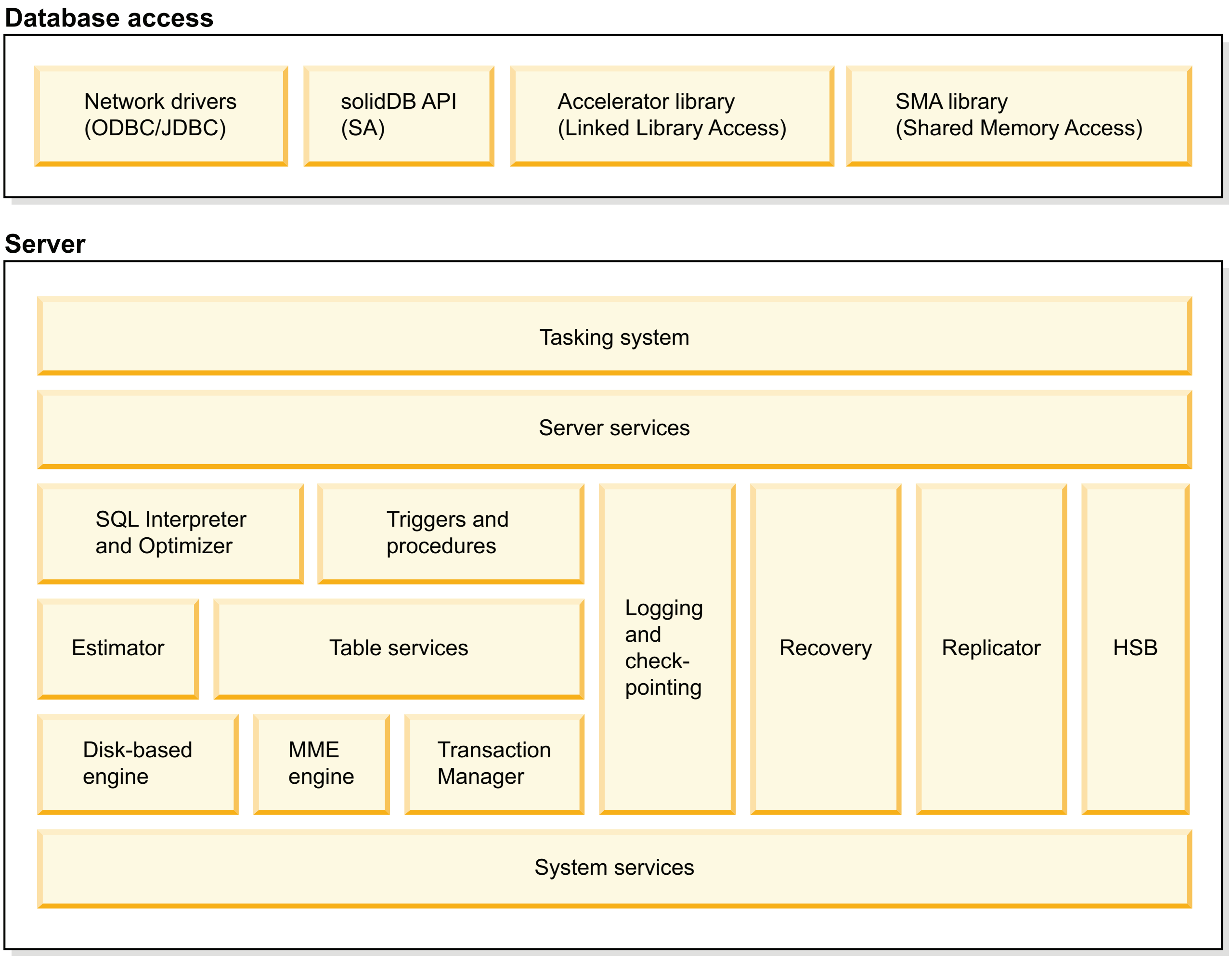


Figure 7. solidDB components

Tasking system

The tasking system is a framework to abstract threads to a concept task. Tasking system implements concurrent execution of the tasks also in single threaded systems.

Server services

The server services component contains services and utilities to use components on the lower levels.

SQL interpreter and optimizer

The SQL interpreter and optimizer is responsible of SQL-clause parsing and optimization. The solidDB server uses SQL syntax that is based on the ANSI X3H2 and IEC/ISO 9075 SQL standards. The SQL-89 Level 2 standard is fully supported and SQL-92 Entry Level. Many features of full SQL-92, SQL-99, and SQL-2003 standards are also supported.

solidDB contains a cost-based optimizer, which ensures that even complex queries can be run efficiently. The optimizer automatically maintains information about table sizes, the number of rows in tables, the available indexes, and the statistical distribution of the index values.

Triggers and procedures

The triggers and procedures component contains a mechanism for parsing and executing SQL-based stored triggers and procedures:

- A trigger activates stored procedure code, which the solidDB server automatically executes when a user attempts to change the data in a table.
- Stored procedures are simple programs, or procedures, that are executed in the solidDB server. You can create procedures that contain several SQL statements or whole transactions, and execute them with a single call statement. In addition to SQL statements, 3GL type control structures can be used enabling procedural control. In this way complex, data-bound transactions can be run on the server itself, thus reducing network traffic.

Logging and checkpointing

The logging and checkpointing component is responsible for maintaining persistency of transactions by write-ahead logging, consistency, and recoverability of the database by checkpointing. Various durability options are available. Reading the transaction log file as it is being written by the server is possible and is done with a special SQL-based interface called *Logreader API*.

Recovery

The recovery component is responsible for recovery from transaction log and database checkpoints.

Replicator

The replicator component provides support for the advanced replication feature. The advanced replication feature is used for asynchronous, pull-based replication between a master database and replica databases. A *master* database contains the master copy of the data. One or more *replica* databases contain full or partial copies of the data. A replica database, like any other database, can contain multiple tables. Some of those tables can contain only replicated data (copied from the master), some can contain local-only data (not copied from the master), and some can contain a mix of replicated data and local-only data. Replicas can submit updates to the master server, which then verifies the updates according to rules set by the application programmers. The verified data is then *published* and made available to all replicas.

HotStandby

The HotStandby (HSB) component enables a secondary server (a hot standby server) to run in parallel with the primary server and keep an up-to-date copy of the data in the primary server.

Estimator

The estimator component provides cost-based estimates for single table access based on projections and constraints. It executes a low-level execution plan generation by using index selection and index range calculations.

Table services

The table services module contains interfaces for single-table access, data type support, transaction management interface, and table and index caches.

Main-memory engine (MME)

The main-memory engine component handles the storage of in-memory tables (M-tables) and indexes.

Transaction manager

The transaction manager component contains commit and rollback implementation and concurrency conflict checking and resolution.

System services

The system services component contains operating system abstraction layer, memory management, thread management, mutexing, and file I/O services.

SQL processing

The solidDB server uses a cost-based optimizer that ensures that SQL statements are executed efficiently.

Optimizer

The solidDB SQL optimizer is a cost-based optimizer. It uses the same techniques as a rules-based optimizer, relying on a preprogrammed set of rules to determine the shortest path to the results. For example, the optimizer considers whether an index exists, if the index is unique, and if the index is over single or composite table columns. However, unlike a rule-based optimizer, the cost-based optimizer can adapt to the actual contents of the database, for example, the number of rows and the value distribution of individual columns.

The server maintains the statistical information about the actual data automatically, ensuring optimal performance. Even when the amount and content of data changes, the optimizer can still determine the most effective route to the data.

Query processing

Queries are processed in small steps to ensure that one time-consuming operation does not block other application requests. A query is processed in a sequence that contains the following phases:

- Syntax analysis
- Creating the execution graph
- Processing the execution graph

Syntax analysis

An SQL query is analyzed and the server produces either a parse tree for the syntax or a syntax error. When a statement is parsed, the information necessary for its execution is loaded into the statement cache. A statement can be executed repeatedly without re-optimization, on condition that its execution information remains in the statement cache.

Creating the execution graph

The execution graph, which contains the following features, is created from the query parse tree.

- Complex statements are written to a uniform and more simple form.
- If better performance can be realized, OR criteria are converted to UNION clauses.
- Intelligent join constraint transfer is performed to produce intermediate join results that reduce the join process execution time.

You can use the EXPLAIN PLAN FOR statement to show the execution plan that the SQL optimizer selected for the SQL statement.

Processing the execution graph

The execution graph is processed in three consecutive phases:

- Type-evaluation phase

The column data types of the result set are derived from the underlying table and view definitions

- Estimate-evaluation phase

  The cost of retrieving first rows and also entire result sets is evaluated, and an appropriate search strategy is dynamically selected based on the parameter values that are bound to the statement.

  The SQL Optimizer bases cost estimates on automatically maintained information about key value distribution, table sizes, and other dynamic statistical data. Manual updates to the index histograms or any other estimation information are not required.

- Row-retrieval phase

  The result rows of the query are retrieved and returned to the client application.

Optimizer hints

Optimizer hints are an extension to solidDB SQL. They are directives that are specified through embedded pseudo comments within query statements. The optimizer detects these directives or hints and bases its query execution plan accordingly. Optimizer hints allow applications to be optimized under various conditions to the data, query type, and the database. They not only provide solutions to performance problems occasionally encountered with queries, but shift control of response times from the system to the user.

solidDB network services

solidDB network services are based on the remote procedure call (RPC) paradigm, which makes the communication interface simple to use. When a client sends a request to the server, it resembles calling a local function. The network services invisibly route the request and its parameters to the server, where the actual service function is called by the RPC server. When the service function completes, the return parameters are sent back to the calling application.

In a distributed system, several applications can request a server to run multiple operations concurrently. For maximum parallelism, solidDB network services use the operating system threads when available to offer a seamless multiuser support. On single-threaded operating systems, the network services extensively use asynchronous operations for the best possible performance.

Communication session layer

The solidDB communication protocol DLLs (or static libraries) offer a standard internal interface to each protocol. The lowest part of the communication session layer works as a wrapper that takes care of choosing the correct protocol DLL or library that relates with the address information. After this point, the actual protocol information of the session is hidden. solidDB can listen to several protocols simultaneously.

Multithread processing

The solidDB multithread architecture provides an efficient way of sharing the processor within an application. A thread is a dispatchable piece of code that merely owns a stack, registers (while the thread is executing), and its priority. It shares everything else with all other active threads in a process. Creating a thread requires much less system overhead than creating a process, which consists of code, data, and other resources such as open files and open queues.

Threads are loaded into memory as part of the calling program; no disk access is therefore necessary when a thread is invoked. Threads can communicate using global variables, events, and semaphores.

If the operating system supports symmetric multithreading between different processors, solidDB automatically takes advantage of the multiple processors.

Types of threads

The solidDB threading system consists of general-purpose threads and dedicated threads.

General-purpose threads

General-purpose threads execute tasks from the server's tasking system. They execute such tasks as serving user requests, making backups, executing timed commands, merging indexes, and making checkpoints (storing consistent data to disk).

General-purpose threads take a task from the tasking system, execute the task step to completion and switch to another task from the tasking system. The tasking system works in a round-robin fashion, distributing the client operations evenly between different threads.

The number of general-purpose threads can be set in the `solid.ini` configuration file.

Dedicated threads

Dedicated threads are dedicated to a specific operation. The following dedicated threadscan exist in the server:

- I/O manager thread

  This thread is used for intelligent disk I/O optimization and load balancing. All I/O requests go through the I/O manager, which determines whether to pass each I/O request to the cache or to schedule it among other I/O requests. I/O requests are ordered by their logical file address. The ordering optimizes the file I/O since the file addresses accessed on the disk are in close range, reducing the disk read head movement.
- Communication read threads

  Applications always connect to a listener session that is running in the selector thread. After the connection is established, a dedicated read thread can be created for each client.
- One communication select thread per protocol (known as the selector thread)

  There is usually one communication selector thread per protocol. Each running selector thread writes incoming requests into a common message queue.
- Communication server thread (also known as the RPC server main thread)

  This thread reads requests from the common message queue and serves applications by calling the requested service functions.

1.3.3 System tools and utilities

The solidDB server package includes console tools for data management and administration, and command-line utilities for data export and import.

The tools and utilities are available in the 'bin' directory in the solidDB server installation directory.

Console tools

solidDB SQL Editor (solsql)
solidDB SQL Editor (**solsql**) is a console tool that you can use to issue SQL statements and solidDB ADMIN COMMANDs at the command prompt. You can also execute script files that contain the SQL statements.

solidDB Remote Control (solcon)
solidDB Remote Control (**solcon**) is a console tool for administration; users with administrator rights can issue ADMIN COMMANDs at the command prompt or by executing a script file that contains the commands. With **solcon**, the ADMIN COMMANDs can be issued as part of the **solcon** startup command line.

Because only users with administrator rights can access **solcon**, if only **solcon** is deployed at a production site, the administrators cannot accidentally execute SQL statements that could change the data.

Tools for exporting and loading data

solidDB Speed Loader (solloado or solload)
solidDB Speed Loader (**solloado** or **solload**) loads data from an external file into a database.

solidDB Export (solexp)
solidDB Export (**solexp**) exports data from a database into a file. It also creates control files used by solidDB Speed Loader (**solloado** or **solload**) to perform data load operations.

solidDB Data Dictionary (soldd)
solidDB Data Dictionary (**soldd**) exports the data dictionary of a database. It produces an SQL script that contains data definition statements that describe the structure of the database.

1.4 Data storage in solidDB

The main-memory engine that handles the in-memory tables (M-tables) and the disk-based engine that handles the storage of disk-based tables (D-tables) use different data storage architectures.

In-memory engine is designed for maximum performance. Knowing that data is always stored in main-memory allows for use of data structures and data access methods that are designed to minimize the computational (CPU) cost of retrieving or updating database records.

Disk-based engine, however, can reduce disk access. Reduced disk access is achieved by data structures and access methods that trade disk access for additional computational processing. Therefore, an in-memory engine has the potential to outperform a disk-based engine even when the latter has cached all the data in the memory buffer pools because it consumes fewer processor cycles to access database records.

1.4.1 Data storage for in-memory tables

The internal part of the server that takes care of storing M-tables is called Main-Memory Engine (MME). In addition to the actual data, the indexes for

M-tables are built in the main memory also. solidDB uses a main-memory-optimized index technology, called *tries*, to implement the indexes.

The basic index structure in the in-memory engine is a *VTrie* (variable length trie) that is optimized variation of the trie. A trie (from retrieval), is a multi-way tree structure that is widely used for storing strings. The idea is that all strings that share a common prefix hang off a common node. For example, when the strings are words over {a..z} alphabet, a node has at most 27 children: one for each letter plus a terminator. VTrie uses bitwise tree where individual bits compose a key allowing keys to be any supported data type. VTrie uses nodes of the capacity of 8 bits. Consequently, each node has at most 257 children, that is, the fan-out is 257 (256 for bits plus a terminator).

A simplified example of the VTrie structure with node capacity of 2 bits and fan-out of four is shown in the following figure.

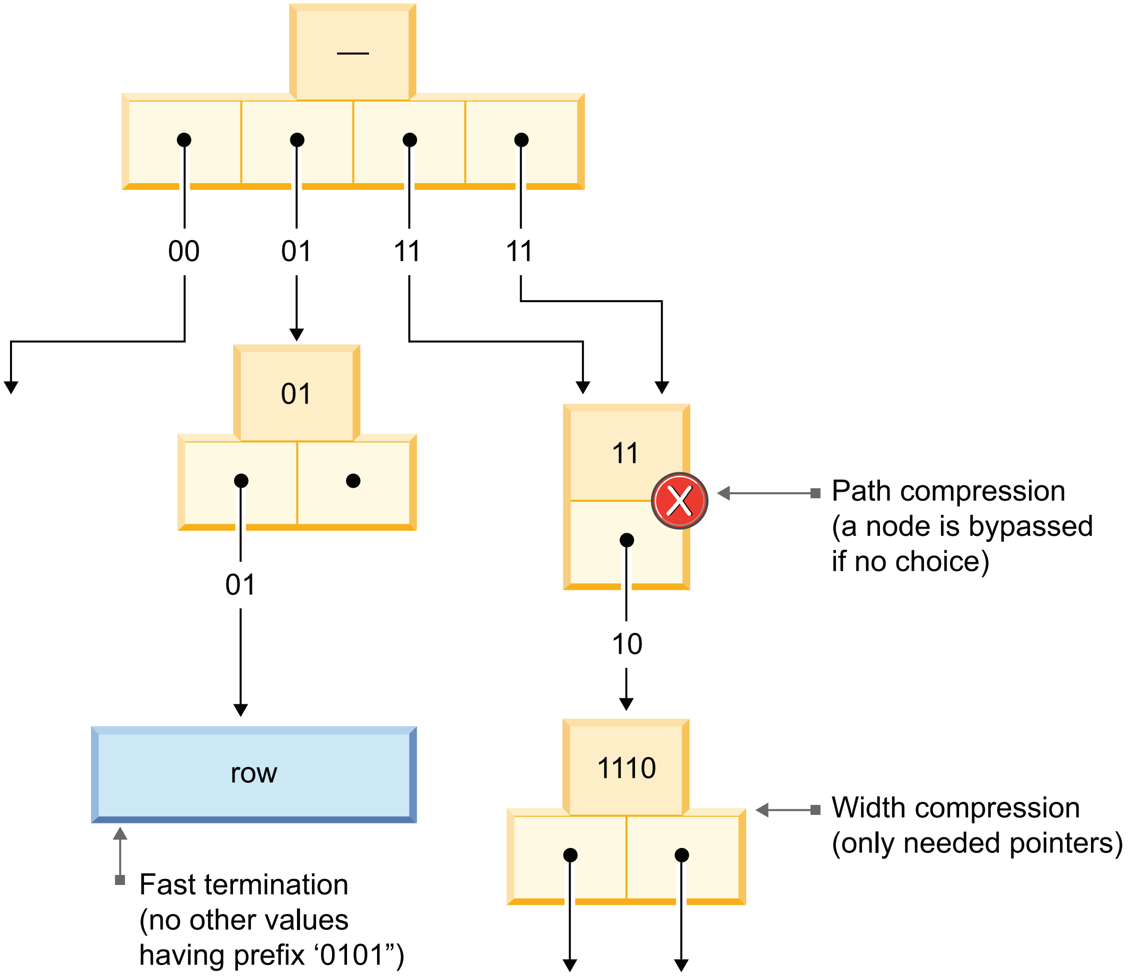


Figure 8. Example of a VTrie structure (simplified)

The elements in a string can be recovered vy using a scan from the root to the leaf nodes that ends a string. All strings in the trie can be recovered by a depth-first browse of the tree.

A competitive solution to VTrie would be a binary search tree. In a binary tree, the node fan-out is two. In each node, you compare a full key value against a node separation value and then choose one of the two children to continue with.

The main advantages of VTries over binary search trees are as follows:

- Looking up keys is faster. Looking up a key of length m takes time that is proportional to m. A binary search tree requires log2(n) comparisons of keys where n is the number of elements in the tree. The total search time is proportional to m log2(n). The advantage of VTrie is because no value comparisons are needed. Each part of a key (a "letter") is applied as an array index to a pointer array of a child node. Contrary to a value comparison, array lookup is a fast operation if the array is cached in processor caches.
- Tries can require less space when they contain a large number of short strings because the keys are not stored explicitly and nodes are shared between keys with common prefix.

Several optimizations are used in Vtrie to speed up retrieval when the key value space is not fully exhausted, as illustrated in Figure 8 on page 19. These are *path compression*, *width compression*, and *fast termination*:

- In path compression, all internal nodes with only one child are removed and a common prefix is stored in the remaining node.
- In width compression, only the needed pointers are stored in the nodes and every node contains a bitmap that stores the information about which pointers are present in the node.
- In fast termination, a pointer to the data record is elevated to a node that represents a prefix that is not shared among the key values.

1.4.2 Data storage for disk-based tables

The internal part of the server that takes care of storing D-tables is called the disk-based engine (DBE). The main data structure that is used to store D-tables is a B-tree variation called *B+tree*. The server uses two structures; the *main storage tree* holds permanent data, and a differential index tree called *Bonsai Tree* stores new data temporarily until it is ready to be moved to the main storage tree.

A basic B+tree with two node levels is illustrated in the following figure.

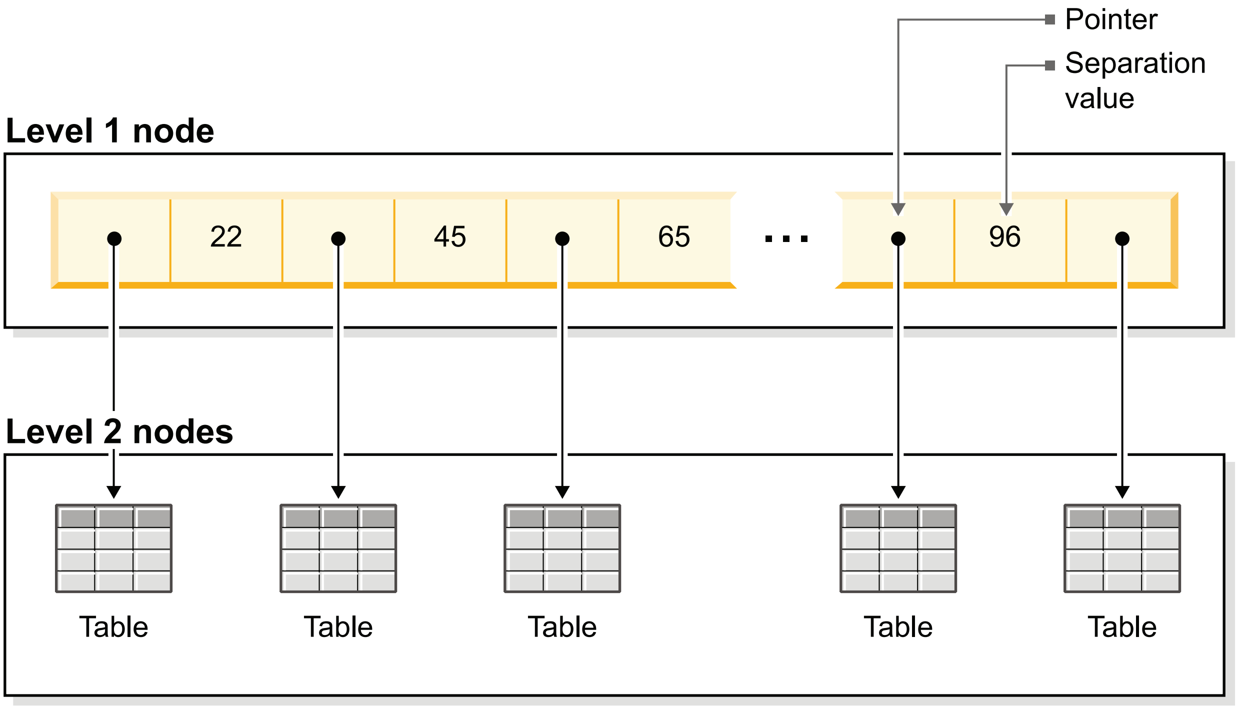


Figure 9. B+tree with two node levels

Each node has a large set of value-pointer pairs. They normally fill a database page that is a unit of data buffering. The page sizes vary from 4 to 32 KB. Compared to VTrie, the page size makes the nodes much larger, resulting in a wide, or bushy tree. The key value is compared against the separation values in the node and, if the key value falls between two separation values, the corresponding pointer is followed to a similar node on the next level. Thanks to a large node size, the number of disk accesses is minimized, and that make B-tree fit for D-tables.

The server uses two incarnations of a B-tree: the *main storage tree* holds permanent data, and a differential index tree called *Bonsai Tree* stores new data temporarily until it is ready to be moved to the main storage tree. In both B-tree structures, two space optimization methods are used. First, only the information that differentiates the key value from the previous key value is saved. The key values are said to be prefix-compressed. Second, in the higher levels of the index tree, the key value borders are truncated from the end; that is, they are suffix-compressed.

Main storage tree

The main storage tree contains all the data in the server, including tables and indexes. Internally, the server stores all data in indexes; there are no separate table stores. Each index contains either complete primary keys (all the data in a row) or secondary keys (what SQL refers to as indexes, which are the column values that are part of the SQL index). There is no separate storage method for data rows, except for binary large objects (BLOB) and other long column values.

All the indexes are stored in a single tree, which is the main storage tree. Within that tree, indexes are separated from each other by a system-defined index identification that is inserted in front of every key value. This mechanism divides the index tree into several logical index subtrees where the key values of one index are clustered close to each other.

Bonsai Tree multiversioning and concurrency control

The Bonsai Tree is a small active index (data storage tree) that efficiently stores new data (deletes, inserts, updates) in central memory, maintaining multiversion information. Multiple versions of a row (old and new) can coexist in the Bonsai Tree. Both the old and new data are used for concurrency control and for ensuring consistent read levels for all transactions without any locking overhead. With the Bonsai Tree, the effort that is needed for concurrency control is reduced.

When a transaction is started, it is given a sequential Transaction Start Number (TSN). The TSN is used as the "read level" of the transaction; all key values that are inserted later into the database from other connections are not visible to searches within the current transaction. This approach offers consistent index read levels that appear as though the read operation was performed at the time the transaction was started. This way ensures that read operations are presented with a consistent view of the data without the need for locks, which have higher overhead.

Old versions of rows (and the newer version or versions of those same rows) are kept in the Bonsai Tree for as long as transactions need to see those old versions. After the completion of all transactions that reference the old versions, the old versions of the data are discarded from the Bonsai Tree, and new committed data is moved from the Bonsai Tree to the main storage tree. The presorted key values are merged as a background operation concurrently with normal database operations. This way offers significant I/O optimization and load balancing. During the merge, the deleted key values are physically removed.

1.5 Application framework support

You can integrate the solidDB database in a number of application frameworks. Application frameworks facilitate application development in Java or C programming languages by abstracting a number of database concepts from the application layer and accessing the database as a generic JDBC or ODBC data source.

Using an application framework can simplify the process of porting an application to use a different database server, because changes are needed only in the database connectivity layer that is managed by the framework, rather than in the application code itself.

An application server provides the infrastructure for executing applications that run your business. It insulates the infrastructure from hardware, operating system, and the network. An application server also serves as a platform to develop and deploy your web services and Enterprise JavaBeans (EJBs), and as a transaction and messaging engine that delivers business logic to users on various client devices. The application server acts as middleware between backend systems and clients. It provides a programming model, an infrastructure framework, and a set of standards for a consistent designed link between them.

Many applications that are written within the application development paradigms can benefit from low database transactional latency and improved database throughput that results from the solidDB in-memory database technology and its ability to bring data close to the application. For example, the SMA functionality can be used within these frameworks on the condition that the solidDB server runs on the same computer as the application server.

WebSphere® Application Server

IBM WebSphere Application Server is the IBM runtime environment for applications that are based on Java. WebSphere Application Server provides the environment to run your solutions and to integrate them with every platform and system as business application services that conform to the service-oriented architecture (SOA) reference architecture. WebSphere Application Server is a key SOA building block. From the SOA perspective, with WebSphere Application Server you can perform the following functions:

- Build and deploy reusable application services quickly and easily
- Run services in a secure, scalable, highly available environment
- Connect software assets and extend their reach
- Manage applications effortlessly
- Grow as your needs evolve, reusing core skills and assets

WebSphere Application Server is available on a wide range of platforms and in multiple packages to meet specific business needs. By providing the application server that is required to run specific applications, it also serves as the base for other WebSphere products, such as IBM WebSphere Enterprise Service Bus, WebSphere Process Server, WebSphere Portal, and many other IBM software products.

More information about using the solidDB server with the WebSphere Application Server is available in the *Configuring WebSphere Application Server with solidDB* article, available on the IBM Software Support portal: http://www.ibm.com/support/docview.wss?uid=swg21406956.

The solidDB data store helper archive for use with WebSphere (`SolidDataStoreHelper.jar`) is included in the solidDB server installation package (`jdbc` directory).

Hibernate

Hibernate is an open source persistence and query framework that provides object-relational mapping of Plain Old Java Objects (POJOs) to relational database tables, and data query and retrieval capabilities. With hibernate, you can write database applications without writing SQL statements.

The mapping between objects and the solidDB database is facilitated with a dialect for Hibernate. The dialect enables the Hibernate library to communicate with the solidDB server. It contains information about the mapping of Java types to SQL types and the functions the solidDB database supports with Hibernate. In general, a Java class maps to a database table, and a Java type maps to an SQL data type. Hibernate eases migration between different databases: you can write an application for a database that in principle works with all databases supported by Hibernate, that is, with any database that provides a dialect.

More information about using the solidDB server with Hibernate is provided in the *Hibernate and solidDB* article, available on the IBM Software Support portal at: http://www.ibm.com/support/docview.wss?uid=swg21440246.

The solidDB dialect for Hibernate (`SolidSQLDialect.jar`) is included in the solidDB server installation package (`jdbc` directory).

JBoss Application Server

JBoss Application Server (or JBoss AS) is an open source application server product based on Java. It was originally developed by JBoss Inc, and is now owned by Red Hat. More information about using the solidDB server with the JBoss Application Server is provided in the *Configuring JBoss Application Server for solidDB* article, available on the IBM Software Support portal at: http://www.ibm.com/support/docview.wss?uid=swg21452681.

WebLogic Application Server

WebLogic Application Server is an application server product, which is owned by the Oracle Corporation and is a part of the Oracle WebLogic Java EE platform product family. More information about using the solidDB server with the WebLogic Application Server is provided in the *Configuring WebLogic Server for solidDB* article, available on the IBM Software Support portal at: http://www.ibm.com/support/docview.wss?uid=swg21439319.

Part 2. Getting started with solidDB

2 Architectural overview of solidDB

The solidDB database server uses a client/server model; the solidDB deployment consists of cooperating server and client processes. The server process manages the database files, accepts connections to the database from client applications, and carries out actions on the database as requested by the clients.

The client process is used to pass the required tasks (through the server process) to the database. There can be several client types: a client can be a command-line tool, a graphical application, or a database maintenance tool. Typically, different applications act as clients to connect to solidDB.

The client and the server can be on different hosts (nodes), in which case they communicate over a network. solidDB provides simultaneous support for multiple network protocols and connection types. Both the database server and the client applications can be simultaneously connected to multiple sites using multiple different network protocols.

solidDB can also run within the application process. This is provided by solidDB *shared memory access* (SMA) and *linked library access* (LLA). In this case, the application is linked to a function library that is provided with the product. The linked application communicates with the server by using direct function calls, thus skipping the overhead that is required when the client and server communicate through network protocols such as the TCP/IP. By replacing the network connection with local function calls, performance is improved significantly.

To submit a query (an SQL statement) to a database server, a client must be able to communicate with that database server. solidDB, like many other database servers, uses *drivers* to enable this communication. Client applications call functions in the driver, and the driver then handles the communications and other details with the server. For example, you might write a C program that calls functions in the (ODBC) driver, or you might write a Java program that calls functions in the (JDBC) driver.

2.1 Installation topologies for solidDB

You can install the solidDB server, drivers, and tools and utilities on a single computer for a simple evaluation topology, or on independent servers for distributed topologies that match your business needs for greater capacity, high availability, embedded deployments, asynchronous and synchronous replication models, and so on.

2.1.1 Example: Evaluation topology of solidDB

In a typical evaluation setup, the solidDB, the ODBC and JDBC drivers, and the solidDB tools are installed on a single computer.

The evaluation topology is appropriate for exploring the software or for training environments, but not for production environments.

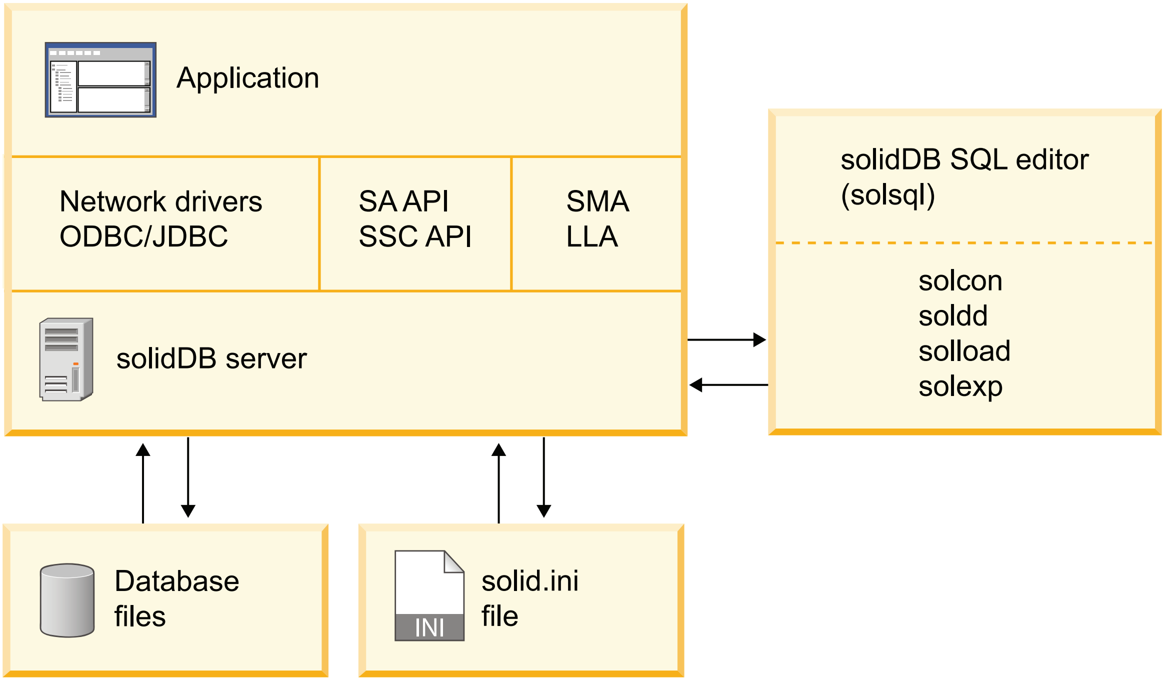


Figure 10. solidDB evaluation topology example

2.1.2 Example: High availability topology

In a typical high availability setup, a two-node, hot-standby configuration is used. The solidDB servers are installed on independent servers.

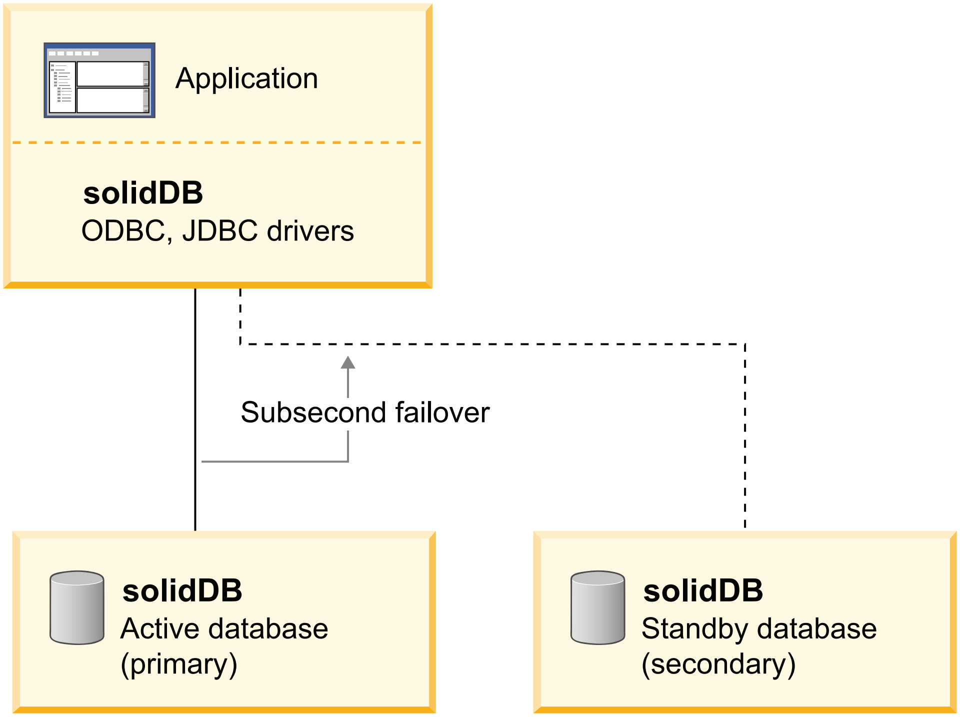


Figure 11. Example: High availability topology

2.1.3 Example: SMA topology

In a shared memory access (SMA) setup, the solidDB server must be installed on the same node as the application. Multiple applications can be installed on the same node.

Other network-connections-based clients and applications on other nodes can also communicate with the SMA server.

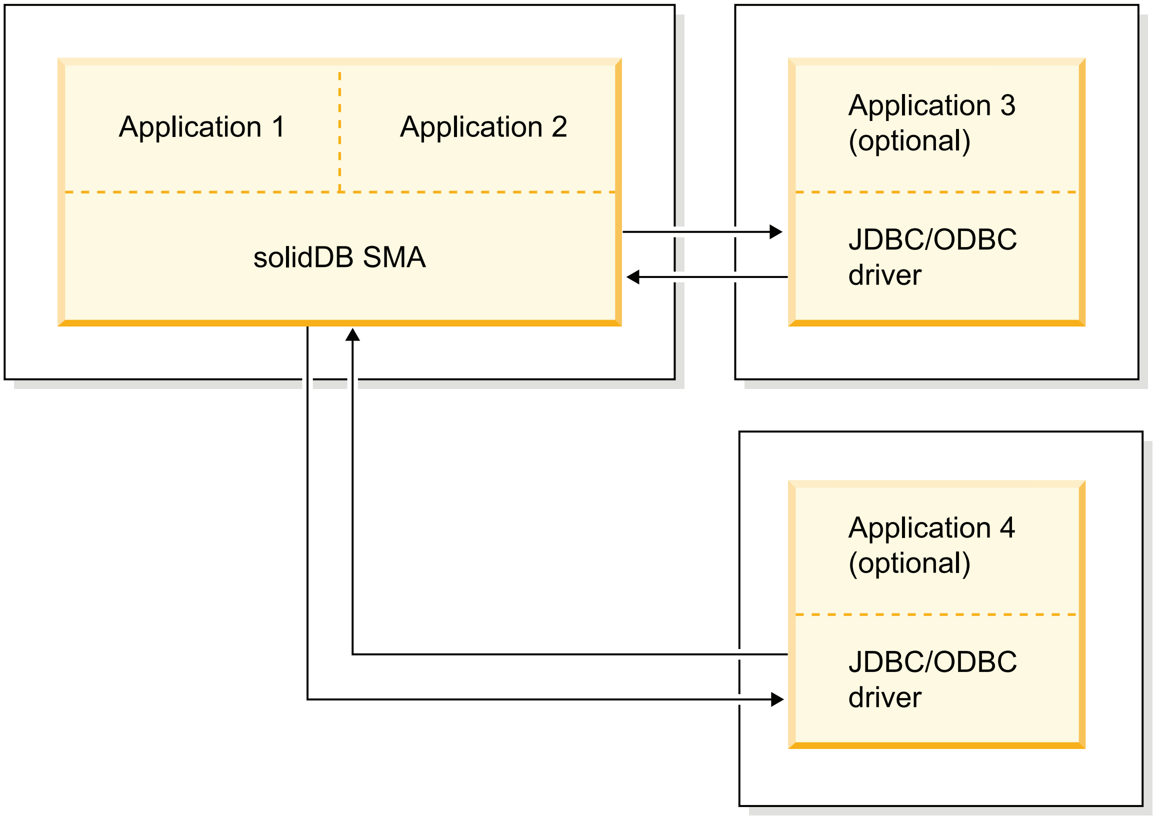


Figure 12. SMA topology example

2.1.4 Example: Advanced replication topology

In a typical advanced replication setup, a multinode configuration is used. The solidDB servers are installed on independent servers. The application can connect to the solidDB master and replica servers using ODBC and JDBC drivers, or SMA or LLA.

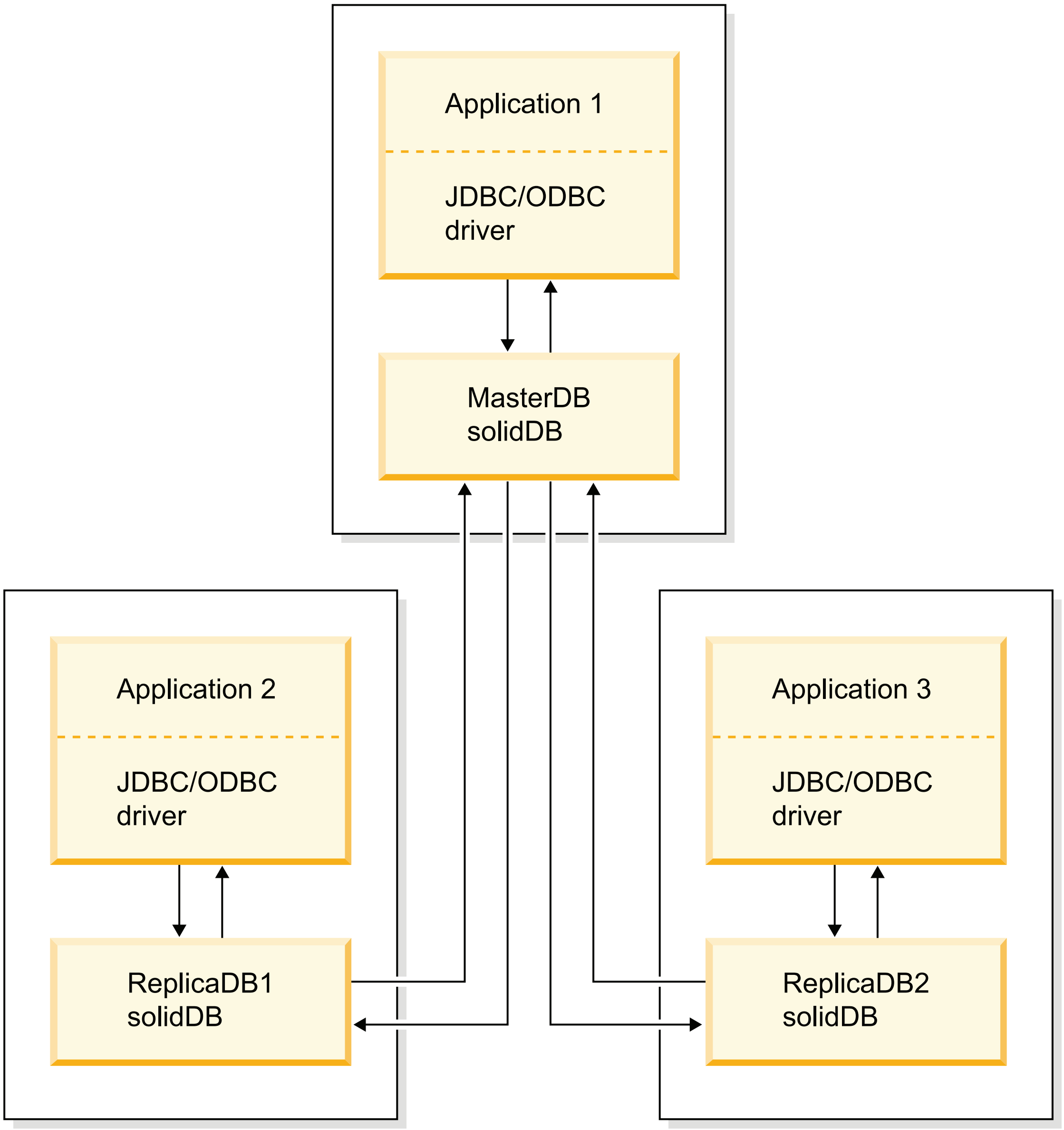


Figure 13. Advance replication installation topology example

3 Product and installation package information

The solidDB V7.0 offering is composed of solidDB and InfoSphere CDC components. The following table shows the installation packages that are included in the solidDB V7.0 offering. Each listed component must be installed separately.

Table 3. solidDB V7.0 installation packages

Component	**solidDB**	**solidDB with InfoSphere CDC replication**	**solidDB with Universal Cache**
IBM solidDB 7.0	X	X	
IBM InfoSphere Change Data Capture solidDB 7.0		X	
IBM InfoSphere Change Data Capture Access Server 6.5		X	
IBM InfoSphere Change Data Capture Management Console 6.5		X	
IBM InfoSphere Change Data Capture *backend data server* 6.5 One of the following: • IBM InfoSphere Change Data Capture DB2® Linux, UNIX, and Windows 6.5 • IBM InfoSphere Change Data Capture Informix® 6.5 • IBM InfoSphere Change Data Capture Microsoft SQL Server 6.5 • IBM InfoSphere Change Data Capture Oracle Trigger 6.5 • IBM InfoSphere Change Data Capture Oracle Redo 6.5 • IBM InfoSphere Change Data Capture Sybase 6.5 • IBM InfoSphere Change Data Capture DB2 z/OS® 6.5 • IBM InfoSphere Change Data Capture DB2 iSeries® 6.1			X
IBM Data Server Driver for ODBC and CLI 9.7 **Note:** Needed only in Universal Cache configurations with SQL passthrough when the backend data server is an IBM data server.			X
IBM solidDB 7.0 License Certificate	X	X	X
IBM solidDB 7.0 Documentation	X	X	X
IBM InfoSphere Change Data Capture Documentation 6.5		X	X

3.1 solidDB server package

The solidDB server package contains a complete set of the server software, including the JDBC and ODBC drivers and various utility programs.

The solidDB server package is delivered with an evaluation license certificate file, `solideval.lic`. With the evaluation license, you can evaluate solidDB for 90 days. For acquiring a permanent license, contact IBM Corporation.

Table 4. solidDB installation images

Component name	Installation package
IBM solidDB 7.0	**Linux and UNIX** solidDB-7.0-<platform>.bin **Windows** solidDB-7.0-<platform>.exe

3.1.1 Directory structure

The default installation of solidDB 7.0 creates a directory called `solidDB7.0`.

The files and subdirectories in the `solidDB7.0` installation directory are explained in the following table.

Table 5. `solidDB7.0` directory structure

Location	Explanation
Root directory	The root directory contains, for example: • a script that is used to facilitate running samples in the database evaluation phase • the evaluation license file • the `welcome.html` file for accessing the package documentation
`bin`	solidDB binary files and dynamic library files
`bin/C` `bin/N`	Auxiliary libraries for IBM Global Security Kit (GSKit)
`doc_html`, `doc_txt`	Package documentation in HTML and text format
`eval_kit/standalone`	Working directory for an evaluation version of the solidDB server. This directory contains a sample `solid.ini` configuration file and an evaluation license file (`solideval.lic`).
`eval_kit/cdc`	Working directory for an evaluation version of the solidDB server for use with Universal Cache or InfoSphere CDC replication. This directory contains a sample `solid.ini` configuration file and an evaluation license file (`solideval.lic`).
`include`	C program headers
`jdbc`	solidDB JDBC Driver Data store helper archive for use with WebSphere (`SolidDataStoreHelper.jar`) solidDB dialect for Hibernate (`SolidSQLDialect.jar`)
`lib`	Static linkable library files

Table 5. `solidDB7.0` directory structure (continued)

Location	Explanation
`lib32`	32-bit static linkable library files – 64-bit AIX® and Solaris packages only The 32-bit libraries can be installed on 64-bit systems. The 64-bit libraries cannot be installed on 32-bit systems.
`licence`	License and notices files
`manuals`	The English versions of the manuals in PDF format can be downloaded to this folder and then accessed through the **Manuals** link on the **Welcome** page
`procedures`	SQL scripts for creating and running stored procedures for data aging and refresh
`properties`	Metadata for IBM Tivoli® Usage and Accounting Manager
`samples`	Samples that can be used in the database evaluation phase and future application development

3.1.2 Library file names

The solidDB server provides many files as linkable libraries.

Most of the library files fall into one of the following categories:

- ODBC drivers
- Shared memory access and linked library access files
- Communication library files
- SA (Server API) library file

All platforms do not have every file. For example, some communication library files are available on Windows environments only.

Some library files are static, that is, they are linked to the client application executable program when you do a compile-and-link operation. Other library files are dynamic: these files are stored separately from your executable program and are loaded into memory when your program runs. For many libraries, the solidDB server provides both a static and a dynamic version on some or all platforms.

Library files are found in the following two directories:

- `bin`
- `lib`

As a rule, the `bin` directory contains dynamic libraries (in addition to executable files), while the `lib` directory contains static libraries. On Windows environments, the `lib` directory also contains the import libraries.

Additionally, on Windows environments, the ODBC and communication `.dll` libraries are copied to the `C:\Windows\system32` directory.

If you use the 32-bit installation program to install the solidDB server on a 64-bit environment, the `.dll` library files are copied to the `C:\Windows\SysWOW64` directory.

The exact library file names depend on the platform. See the following tables for examples on Windows and Linux environments:

Table 6. Example: solidDB library files in Windows 32-bit package

File name	**Description**
bin\	
sacw3270.dll	ODBC library - ASCII
snpw3270.dll	NamedPipes communication protocol link library
socw3270.dll	ODBC library - Unicode
sosw3270.dll	ODBC Driver Manager setup library
ssaw3270.dll	solidDB SA API library
ssolidac70.dll	Linked library access (LLA) dynamic library
stcw3270.dll	TCP/IP communication protocol link library
lib\	
solidctrlstub.lib	solidDB Control API (SSC) stub library.
	This static library is used if you want to write code that can be run either locally with the linked library access library, or remotely without the linked library access.
solidimpac.lib	Linked library access (LLA) import library
solidimpodbca.lib	ODBC import library - ASCII
solidimpodbcu.lib	ODBC import library - Unicode
solidimpsa.lib	solidDB SA API import library

Table 7. Example: solidDB library files in Linux 32-bit package

File name	**Description**
bin\	
sacl2x70.so	ODBC shared library - ASCII
socl2x70.so	ODBC shared library - Unicode
ssal2x70.so	solidDB SA API library
ssolidac70.so	Linked library access (LLA) shared library
ssolidsma70.so	Shared memory access (SMA) shared library
lib\	
solidctrlstub.a	solidDB Control API (SSC) stub library.
	This static library is used if you want to write code that can be run either locally with the linked library access library, or remotely without the linked library access.
solidac.a	Linked library access (LLA) static library
solidodbca.a	ODBC static library - ASCII
solidodbcu.a	ODBC static library - Unicode
solidsa.a	solidDB SA API static library
libssolidac70.so	Symbolic link for shared LLA library
libssolidsma70.so	Symbolic link for shared SMA library

Table 7. Example: solidDB library files in Linux 32-bit package (continued)

File name	Description
libsacl2x70.so	Symbolic link for shared ODBC library - ASCII
libsocl2x70.so	Symbolic link for shared ODBC library - Unicode
libssal2x70.so	Symbolic link for shared solidDB SA API library
libsolidodbca.a	Symbolic link for static ODBC library - ASCII
libsolidodbcu.a	Symbolic link for static ODBC library - Unicode
libsolidsa.a	Symbolic link for static solidDB SA API library
libsolidac.a	Symbolic link for static LLA library

For a list of the library file names on your installation of solidDB server, see the SDK Notes in the solidDB package, accessible through the **Welcome** page in your solidDB installation directory.

Dynamic library file naming conventions

Dynamic library files use the following naming convention:

```
sLLpppVV.eee
```

where

- `LL` = purpose of the library
  - `ac`: ODBC library - ASCII
  - `np`: NamedPipes communication protocol link library
  - `oc`: ODBC library - Unicode
  - `os`: ODBC Driver Manager setup (for Windows only)
  - `sa`: solidDB SA API library
  - `solidac`: Linked library access (LLA) dynamic library
  - `solidsma`: Shared memory access (SMA) dynamic library
  - `tc`: TCP/IP communication protocol link library
- `ppp` = platform
  - `a5x64`: AIX, 64-bit
  - `hia64`: HP-UX 11 64-bit (IA64)
  - `l2x`: Linux for x86
  - `l2x64`: Linux for x86, 64-bit
  - `lzx64`: Linux for System z, 64-bit
  - `s0x64`: Solaris 10 (SPARC, 64-bit)
  - `s0xi64`: Solaris 10 (ix86, 64-bit)
  - `w32`: Windows 32-bit (x86)
  - `w64`: Windows 64-bit (x86)
- `VV` = first two digits of the solidDB version, for example 70 for version 7.0, 63 for version 6.3
- `eee` = platform-specific file name extension:
  - `*.dll` Dynamic Link Library for Windows
  - `*.so` (Shared Object) for AIX, HP-UX, Linux, and Solaris

3.1.3 ODBC, JDBC, and proprietary programming interfaces

The solidDB server provides ODBC, JDBC and proprietary interfaces for clients.

For more details, see the *IBM solidDB Programmer Guide*.

solidDB JDBC Driver 2.0

Table 8. solidDB JDBC Driver 2.0 key information

Compatibility	JDBC 2.0, with selected features of JDBC 2.0 Optional Package
Driver location	`<solidDB installation directory>/jdbc/SolidDriver2.0.jar`

JDBC URL format	`jdbc:solid://<hostname>:<port>/<username>/<password>[?<property-name>=<value>]...`
	For example:
	`"jdbc:solid://localhost:1964/dba/dba"`

Driver class name	solid.jdbc.SolidDriver

Standard compliance

The solidDB JDBC 2.0 Driver supports the JDBC 2.0 specification. Additionally, Connection Pooling, JNDI Data Sources, and Rowsets of the JDBC 2.0 Optional Package (known before as Standard Extension) are supported too.

Non-standard features include support for IBM WebSphere and timeout control extensions.

The following features of the Optional Package are currently supported by the solidDB JDBC 2.0 driver:

- Connection pooling (class solid.jdbc.ConnectionPoolDataSource)
- Connected RowSet (class solid.jdbc.rowset.SolidJDBCRowSet)
- Implemented JDBC data sources:
  - solid.jdbc.DataSource (implements javax.sqlDataSource)
  - solid.jdbc.SolidConnectionPoolDataSource (implements javax.sql.ConnectionPoolDataSource)
- JTA (Java Transaction API) – XA interface for Java (implements javax.transaction.xa.XAResource and javax.transaction.xa.Xid)

Full documentation for the solidDB JDBC Driver is included in the *IBM solidDB Programmer Guide*.

solidDB JDBC Driver extensions

The solidDB JDBC Driver supports the following non-standard extensions. For more information, see the *IBM solidDB Programmer Guide*.

JDBC URL format

You can set the connection property values in the URL string.

Connection timeout

Connection timeout refers to the response timeout of any JDBC call that invokes data transmission over a connection socket. If the response message is not received within the time that is specified, an I/O exception is thrown. The JDBC standard (2.0/3.0) does not support setting of the connection timeout. The solidDB product has two ways for doing that: one using a non-standard driver manager extension method and the other using the property mechanisms. The time unit in either case is 1 ms.

Login timeout

The timeout fires at the connect time. The setting is implemented with a connection property. The property overrides the login timeout for JDBC specified by other means (like login timeout parameter in Driver Manager).

Connection idle timeout

If the connection is inactive for the amount of time specified with the idle timeout property, the server closes the connection. The connection idle timeout property overrides the server parameter setting for the session.

Statement cache

You can set the size of the statement cache for a connection.

Transparent Connectivity Support

solidDB JDBC driver fully supports solidDB Transparent Connectivity (TC) including transparent failover and load balancing. See the *IBM solidDB High Availability User Guide* for more information about usage of Transparent Connectivity.

Shared memory access (SMA) connection property

The SMA connection property defines that the driver connects to a SMA server with a local connection, bypassing network protocols.

SQL passthrough connection properties

The SQL passthrough connection property defines the default passthrough mode for the connection.

Catalog and schema name connection properties

You can set the catalog and schema names for the connection.

WebSphere support

To support WebSphere, adata source adapter SolidDataStoreHelper is provided in a separate file `SolidDataStoreHelper.jar`, in the `'jdbc'` directory of thesolidDB package.

solidDB ODBC Driver 3.5.x

solidDB provides two ODBC drivers, one for Unicode and one for ASCII character sets. For more information about these drivers, see the *IBM solidDB Programmer Guide*.

The following functions are not supported:

- SQLBrowseConnect
- SQLSetScrollOptions
- SQLParamOptions
- SQLNativeSql
- SQLMoreResults

ODBC extensions

solidDB ODBC driver incorporates several extensions for, for example, timeout controls, statement cache behavior, and support for Transparent Connectivity. For more information, see the *IBM solidDB Programmer Guide*.

Proprietary interfaces

The solidDB Application Programming Interface (SA API) and solidDB Server Control API (SSC API) allow, for example, C programs to directly call functions inside the database server. These proprietary interfaces are provided with the solidDB shared memory access (SMA) and linked library access (LLA) libraries.

3.1.4 System tools and utilities

The solidDB server package includes console tools for data management and administration, and command-line utilities for data export and import.

The tools and utilities are available in the 'bin' directory in the solidDB server installation directory.

Console tools

solidDB SQL Editor (solsql)
solidDB SQL Editor (**solsql**) is a console tool that you can use to issue SQL statements and solidDB ADMIN COMMANDs at the command prompt. You can also execute script files that contain the SQL statements.

solidDB Remote Control (solcon)
solidDB Remote Control (**solcon**) is a console tool for administration; users with administrator rights can issue ADMIN COMMANDs at the command prompt or by executing a script file that contains the commands. With **solcon**, the ADMIN COMMANDs can be issued as part of the **solcon** startup command line.

Because only users with administrator rights can access **solcon**, if only **solcon** is deployed at a production site, the administrators cannot accidentally execute SQL statements that could change the data.

Tools for exporting and loading data

solidDB Speed Loader (solloado or solload)
solidDB Speed Loader (**solloado** or **solload**) loads data from an external file into a database.

solidDB Export (solexp)
solidDB Export (**solexp**) exports data from a database into a file. It also creates control files used by solidDB Speed Loader (**solloado** or **solload**) to perform data load operations.

solidDB Data Dictionary (soldd)
solidDB Data Dictionary (**soldd**) exports the data dictionary of a database. It produces an SQL script that contains data definition statements that describe the structure of the database.

3.1.5 Samples

The solidDB package includes several sample programs that are written in C, SQL, and Java. The samples help you get started with the key features of solidDB.

The samples and scripts are located in the 'samples' directory below the solidDB installation directory. Each sample directory includes a readme.txt file that provides instructions for how to use the samples.

3.2 InfoSphere CDC packages

The InfoSphere CDC components are delivered as separately deployable packages.

3.2.1 InfoSphere CDC for solidDB

The InfoSphere CDC for solidDB package contains the software for the replication engine that captures and transfers data changes between solidDB and other databases.

Table 9. InfoSphere CDC for solidDB installation images

Component name	Installation package	Contents

InfoSphere Change Data Capture solidDB	**Linux and UNIX**
	`setup-cdc-<platform>-solid.bin`
	For example:
	`setup-cdc-linux-x86-solid.bin`
	Windows
	`setup-cdc-x86-solid.exe`
		• Software for the configuration tool and the InfoSphere CDC instance for solidDB
		• solidDB JDBC Driver (`SolidDriver2.0.jar` in the `/lib` directory)
		• Tools, utilities, and samples (`/samples` directory)
		– Automation tools, utilities, and samples for scripting most common InfoSphere CDC tasks (`ucutils`, `ucpassthrough`, and `uchsbmonitor` directories)
		– Generic InfoSphere CDC samples for Java user exits and SQL scripts
		• InfoSphere CDC API documentation (`/docs` directory)

3.2.2 InfoSphere CDC Access Server

The InfoSphere CDC Access Server package contains the software for controlling access to the replication environment.

Table 10. InfoSphere CDC for Access Server installation images

Component name	Installation package	Contents

InfoSphere Change Data Capture Access Server	**Linux and UNIX**
	`cdcaccess-<version>-setup.bin`
	For example:
	`cdcaccess-6.5.1618.0-solaris-sparc-setup.bin`
	Windows
	`cdcaccess-<version>-setup.exe`
	For example:
	`cdcaccess-6.5.1618.0-setup.exe`
		• Software for controlling access to your replication environment

3.2.3 InfoSphere CDC Management Console

The InfoSphere CDC Management Console package contains the software for configuring and monitoring user access and replication subscriptions. Management Console is available only on Windows environments.

Table 11. InfoSphere CDC Management Console installation images

Component name	Installation package	Contents
InfoSphere Change Data Capture Management Console	**Linux and UNIX** Not applicable, Management Console is available only on Windows environments **Windows** `cdcmc-<version>-setup.exe` For example:`cdcmc-6.5.1618.0-setup.exe`	• Software for configuring and monitoring InfoSphere CDC user access and replication subscriptions • PDF format *InfoSphere Change Data Capture Management Console, Administration Guide* (`/documentation` directory) • Online help (accessible through **Help** menu in the Management Console user interface) • IBM Java SDK and Runtime Environment Guides (`/docs` directory)

3.3 Documentation packages

Documentation for solidDB is composed of the *IBM solidDB Documentation* package and the *InfoSphere Change Data Capture Documentation* package. Both packages are available as an online information center and in PDF format.

3.3.1 solidDB documentation

solidDB documentation is available online in the solidDB 7.0 Information Center and in PDF format. Most up-to-date information is always available in the Information Center.

Delivery of solidDB documentation

solidDB 7.0 Information Center

The most up-to-date solidDB documentation is available in the information center format at http://publib.boulder.ibm.com/infocenter/soliddb/v7r0/.

solidDB manuals in PDF format

The PDF manuals are available for download at the following locations:

- Software Support portal for solidDB at ftp://ftp.software.ibm.com/software/data/soliddb/info/7.0/man/.
- IBM Publications Center at http://www.elink.ibmlink.ibm.com/publications/servlet/pbi.wss

In addition, the PDF format manuals are available as the *IBM solidDB Documentation* package. This package is delivered together with the software packages in IBM Passport Advantage®, or in the Quick Start DVD in physical media deliveries.

Tip: If you download the English version PDF files to the `manuals` directory in your solidDB server installation directory, you can access the manuals also through the **Welcome** page of your solidDB software package. For detailed instructions, see section Installing solidDB Documentation package.

3.3.2 InfoSphere CDC documentation

InfoSphere CDC for solidDB documentation is included in the *IBM solidDB Documentation* package. Documentation for InfoSphere CDC Management Console, InfoSphere CDC Access Server, and InfoSphere CDC engine for the back-end data server is part of the *InfoSphere Change Data Capture Documentation* package.

Delivery and location of documentation for InfoSphere CDC components

The *InfoSphere Change Data Capture Documentation* package is available in information center and PDF format:

- IBM InfoSphere Change Data Capture version 6.5 Information Center
- InfoSphere Change Data Capture 6.5 End-User Documentation in PDF format - IBM Software Support Portal
- Embedded Help accessible through the Management Console **Help** menu
- *InfoSphere Change Data Capture Documentation* installation package (PDF format), available at Passport Advantage

3.4 License certificate

The `solid.lic` license file is distributed as a separate download image called the License Certificate.

The part number for IBM solidDB V7.0 License Certificate is **CI3ZTEN**. The License Certificate is available for download at IBM Passport Advantage. In physical media deliveries, the License Certificate is included in the Quick Start DVD.

Downloading solidDB license from IBM Passport Advantage

1. Sign in to IBM Passport Advantage with your IBM ID.
2. Select the **Software download & media access** link.
3. On the **Download finder** tab, select **Find by description or part number**.
4. Search for the IBM solidDB V7.0 License Certificate with the part number **CI3ZTEN**.

After you have downloaded the license certificate, unarchive the download image and copy the `solid.lic` file to your solidDB working directory or the location that is defined with the SOLIDDIR environment variable.

4 Installation of solidDB server

The following sections provide instructions for installing solidDB server and the solidDB Documentation package.

4.1 System requirements

IBM solidDB supports more than 30 different platforms, each understood as a combination of hardware type and operating system. Typically all the current commonly used platforms are supported, as well as some legacy platforms.

4.1.1 IBM solidDB supported platforms

The following table shows an overview of the supported platforms for the components included in the IBM solidDB 7.0 product offering.

More detailed information about the platform support for each component is available through the Software product compatibility reports portal on ibm.com®
(see direct links after the table).

Table 12. IBM solidDB supported platforms

Operating system		Hardware	solidDB server 7.0	InfoSphere CDC 6.5											ODBC
				InfoSphere CDC solidDB 7.0	MC	AS	DB2	DB2 z/OS	DB2 iSeries	IDS	OR	OT	MS SQL	Sybase	
AIX	AIX 7.1 AIX 6.1	64-bit systems with POWER5, POWER6®, or POWER7®	X	X		X	X			X	X			X	X
HP-UX	HP-UX 11i v3	Itanium-based HP Integrity Series systems	X	X		X				X	X			X	X
Linux	Red Hat Enterprise Linux (RHEL) 6, 5 SUSE Linux Enterprise Server (SLES) 11, 10	32-bit and 64-bit systems based on Intel or AMD processors that are capable of running the supported Linux operating systems (x86 and x64 systems)	X	X		X	X			X	X			X	X
	Red Hat Enterprise Linux (RHEL) 5 System z SUSE Linux Enterprise Server (SLES) 10 System z	System z	X[1]				X					X			X
Solaris	Solaris 11	64-bit systems with UltraSPARC processors	X[3]												
		64-bit systems with x86 processors	X[3]												
	Solaris 10	64-bit systems with UltraSPARC processors	X	X		X	X			X	X			X	X
		64-bit systems with x86 processors	X												X

Table 12. IBM solidDB supported platforms (continued)

Operating system		Hardware	solidDB server 7.0	InfoSphere CDC 6.5											ODBC
Windows	Windows Server 2012 (Standard Server, Enterprise Server, and Datacenter Editions) Windows 8 (Professional, Enterprise, and Ultimate editions)	32-bit and 64-bit systems based on Intel or AMD processors that are capable of running the supported Windows operating systems (x86 and x64 systems)	X[2]												
	Windows Server 2008 R2, 2008 (Standard Server, Enterprise Server, and Datacenter Editions) Windows 7 (Professional, Enterprise, and Ultimate editions) Windows Vista (Business, Enterprise, and Ultimate editions)		X	X	X	X	X			X			X	X	X
IBM i	i5/OS™ 7.1 i5/OS 6.1 i5/OS 5.4 i5/OS 5.3	POWER® System with i5 processors							X						X
z/OS	z/OS V1.11 z/OS V1.10	System z						X							X

MC = InfoSphere Change Data Capture Management Console 6.5

AS = InfoSphere Change Data Capture Access Server 6.5

DB2 = InfoSphere Change Data Capture DB2 Linux, UNIX, and Windows 6.5

DB2 z/OS = InfoSphere Change Data Capture DB2 z/OS 6.5

DB2 iSeries = InfoSphere Change Data Capture DB2 iSeries 6.1

IDS = InfoSphere Change Data Capture Informix 6.5

OR = InfoSphere Change Data Capture Oracle Redo 6.5

OT = InfoSphere Change Data Capture Oracle Trigger 6.5

MS SQL = InfoSphere Change Data Capture Microsoft SQL Server 6.5

Sybase = InfoSphere Change Data Capture Sybase 6.5

ODBC = IBM Data Server Driver for ODBC and CLI 9.7

[1] Support for System z introduced in V7.0 Fix Pack 1

[2] Support for Windows 8 and Windows Server 2012 introduced in V7.0 Fix Pack 4

[3] Support for Solaris 11 introduced in V7.0 Fix Pack 6

Software product compatibility reports on ibm.com

The Software product compatibility reports portal on ibm.com provides various tools for generating reports on the hardware and software support level of IBM products. Use the following links to view reports specific to IBM solidDB 7.0.

- Operating systems for IBM solidDB 7.0
- IBM solidDB 7.0 on AIX
- IBM solidDB 7.0 on HP-UX
- IBM solidDB 7.0 on Linux
- IBM solidDB 7.0 on Solaris
- IBM solidDB 7.0 on Windows

Related concepts:

4.1.2, "Additional solidDB server installation requirements"

4.1.2 Additional solidDB server installation requirements

Before you install solidDB server, ensure that the system you choose meets the following software and disk and memory requirements.

- About 48 MB of disk space, including the space for separately installed documentation – the number varies considerably, depending on the platform
- At least 40 MB of RAM in the default configuration
- Adequate disk space for your database – an empty database typically requires about 16 MB of disk space
- If you use in-memory tables, additional memory to store those tables
- If you use InfoSphere CDC technology (or, the solidDB log reader is enabled), enough disk space to accommodate transaction log files preserved for replication recovery (catchup) – by default, the required log retention space is 10 GB
- Java Runtime Environment (JRE) or Java Development Kit (JDK), version 1.4.2 or newer, is required for
  - solidDB installation program

    Note: On Linux systems, the installation program does not support GNU Compiler for Java (GCJ).
  - Shared memory access (SMA) and linked library access (LLA) with Java

User process resource limits (`ulimits`) considerations in Linux and UNIX environments

In Linux and UNIX environments, you might need to modify the settings for the user process resource limits (**`ulimits`**) of your system. For details, see *OS user limit requirements (Linux and UNIX)*.

Security-enhanced Linux considerations

On Red Hat Enterprise Linux (RHEL) operating systems, if Security-enhanced Linux (SELinux) is enabled and in enforcing mode, the installer might fail because of SELinux restrictions.

To determine whether SELinux is installed and in enforcing mode, complete one of the following actions:

- Check the `/etc/sysconfig/selinux` file.
- Run the **`sestatus`** command.
- Check the `/var/log/messages` file for SELinux notices.

To disable SELinux, complete one of the following actions:

- Set SELinux in permissive mode and run the **`setenforce 0`** command as a superuser.
- Modify `/etc/sysconfig/selinux` and restart the computer.

If the solidDB server installs successfully on an RHEL system, all solidDB processes will run in the unconfined domain. To assign the processes to their own domains, so that also confined users can run them, you must modify the policy modules.

OS user limit requirements (Linux and UNIX)

In Linux and UNIX environments, you might have to modify the settings for the user process resource limits (**ulimit**) of your system.

If required **ulimit** values are not met, the solidDB server can encounter unexpected operating system resource shortage errors, such as `SOLID Communication Error 21309: Failed to accept a new client connection, out of TCP/IP resources`.

To set the resource limits permanently on your system, modify the hard operating system **ulimit** for the **data**, **nofiles**, and **fsize** resources (root user or a System Administrator rights required).

Table 13. Recommended ulimit values

Hard **ulimit** resource	Description	Recommended value
data	Maximum private memory allowed for a process	Unlimited
nofiles	Maximum number of open files/file descriptors allowed for a process (related to sockets available to the operating system and applications)	65536 or unlimited
fsize	Maximum file size allowed	Unlimited

For instructions on how to query and modify the ulimit values, see your operating system documentation.

Example

To query the hard **ulimit** values:

```
ulimit -Ha
-t: cpu time (seconds) unlimited
-f: file size (blocks) unlimited
-d: data seg size (kbytes) unlimited
-s: stack size (kbytes) 100000
-c: core file size (blocks) unlimited
-n: file descriptors 2048
-v: virtual memory size (kb) unlimited
```

To set the hard **nofiles** value to 65536:

```
ulimit -Hn 65536
```

4.2 Installing solidDB

solidDB can be installed by using the interactive installer for graphical user interface or command line, or by using a silent installation method.

4.2.1 Graphical interface installation

The GUI installation provides a graphical method for installing solidDB.

Before you begin

In Windows environments, you must have Administrator rights to install solidDB.

Procedure

1. Install Java Runtime Environment (JRE) or Java Development Kit (JDK), version 1.4.2 or newer, if not already installed.

   JRE or JDK 1.4.2 or newer is needed to run the solidDB installer.

   Note: On Linux systems, GNU Compiler for Java (GCJ) is not supported.
2. On the downloaded installation image or the installation DVD, locate the installation program file for your operating system:
   - `solidDB-7.0-<platform>.exe` (Windows)
   - `solidDB-7.0-<platform>.bin` (Linux and UNIX)
3. Double-click the installation program file. The solidDB installation wizard starts.
4. Follow the instructions on the wizard to complete the installation.

   Note: In Linux and UNIX operating systems, you must be able write to the directory that you are using for the installation. If the installation program cannot create the directory, you are prompted to specify a different directory.

Results

You have installed solidDB. To see the solidDB package documentation, open `welcome.html` in the installation directory. Alternatively, you can access the ASCII-formatted files in the 'doc_text' directory.

What to do next

- For information about errors that are encountered during installation, review the installation log file (`IBM_solidDB_7.0_InstallLog.log`) located in the installation root directory.
- See also section 4.2.4, “Post-installation tasks,” on page 51.

4.2.2 Console installation

Use the console installation method to install solidDB from a command-line interface.

Before you begin

In Windows environments, you must have Administrator rights to install solidDB.

Procedure

1. Install Java Runtime Environment (JRE) or Java Development Kit (JDK), version 1.4.2 or newer, if not already installed.

   JRE or JDK 1.4.2 or newer is needed to run the solidDB installer.

   Note: On Linux systems, GNU Compiler for Java (GCJ) is not supported.
2. On the downloaded installation image or the installation DVD, locate the installation program file for your operating system:
   - `solidDB-7.0-<platform>.exe` (Windows)
   - `solidDB-7.0-<platform>.bin` (Linux and UNIX)
3. Start the installation program from the command line. Use the following command:
   - Windows

```
<installation_program> -i console
```

For example, in Windows 32-bit operating system:

```
solidDB-7.0-w32.exe -i console
```

- Linux and UNIX

```
sh <installation_program> -i console
```

For example, in Linux 64-bit operating system:

```
sh solidDB-7.0-linux-x86_64.bin -i console
```

4. Follow the displayed instructions to complete the installation.

   Note: In Linux and UNIX operating systems, you must be able write to the directory that you are using for the installation. If the installation program cannot create the directory, you are prompted to specify a different directory.

Results

You have installed solidDB. To see the package documentation, open `welcome.html` in the installation directory. Alternatively, you can access the ASCII-formatted files in the 'doc_text' directory.

What to do next

- For information about errors that are encountered during installation, review the installation log file (`IBM_solidDB_7.0_InstallLog.log`) located in the installation root directory.
- See also section 4.2.4, "Post-installation tasks," on page 51.

4.2.3 Silent installation

Use the silent installation method to install solidDB without any user interaction. This method can be used, for example, for large-scale deployments of solidDB where the silent installation command is embedded in a script.

Before you begin

In Windows environments, you must have Administrator rights to install solidDB.

Procedure

1. Install Java Runtime Environment (JRE) or Java Development Kit (JDK), version 1.4.2 or newer, if not already installed, on all the systems you will be installing solidDB

   JRE or JDK 1.4.2 or newer is needed to run the solidDB installer.

   Note: On Linux systems, GNU Compiler for Java (GCJ) is not supported.
2. Copy the installation program file from the downloaded installation image or the installation DVD to a temporary directory on your system. Use the installation program for your operating system:
   - `solidDB-7.0-<platform>.exe` (Windows)
   - `solidDB-7.0-<platform>.bin` (Linux and UNIX)
3. Start the installation program from the command line and generate a response file with the following command:
   - Windows

     ```
     <installation_program> -r <response-file>
     ```

     For example, in Windows 32-bit operating system:

```
soliddb-7.0—w32.exe -r response.txt
```

- Linux and UNIX

  ```
  sh <installation_program> -r <response-file>
  ```

  For example, in Linux 64-bit operating system:

  ```
  sh solidDB-7.0-linux-x86_64.bin -r response.txt
  ```

The solidDB installation wizard starts.

4. Follow the displayed instructions to complete the installation.

   Note: In Linux and UNIX operating systems, you must be able write to the directory that you are using for the installation. If the installation program cannot create the directory, you are prompted to specify a different directory.
5. After the installation has finished, check that the response file you generated contains the options you need. The response file is located in the installation root directory.

   Tip: A sample response file that is called `installer-response-file.txt` is available in the solidDB installation directory.
6. On another system, perform the silent installation by running the following command:
   - Windows

     ```
     <installation_program> -i silent -f <response-file>
     ```

     For example, in Windows 32-bit operating system:

     ```
     soliddb-7.0—w32.exe -i silent -f response.txt
     ```
   - Linux and UNIX

     ```
     sh <installation_program> -i silent -f <response-file>
     ```

     For example, in Linux 64-bit operating system:

     ```
     sh solidDB-7.0-linux-x86_64.bin -i silent -f response.txt
     ```

4.2.4 Post-installation tasks

After installing the solidDB server, you might want to apply the full product license key, check the solidDB JDBC Driver settings, and enable the use of samples.

Procedure

1. Copy the license file (`solid.lic`) from the License Certificate image to the solidDB working directory.

   Tip: The default installation of solidDB includes an evaluation license (`solideval.lic`) in the installation directory. The evaluation license enables you to evaluate solidDB for 90 days.
2. If you are planning to use the solidDB JDBC Driver, set the CLASSPATH environment variable for your environment to include the solidDB JDBC Driver .jar file installation path. The solidDB JDBC Driver (`SolidDriver2.0.jar`) is located in the `'jdbc'` directory under the solidDB installation directory.
   - **Windows**

     The installation adds the default solidDB JDBC Driver installation path to the system class path environment variable automatically.

     You can check and set the system class path environment variable through the Control Panel:

     Control Panel > **System** > **Advanced** > **Environment Variables**
   - **Linux and UNIX**

Set the classpath environment variable to include the solidDB JDBC Driver (SolidDriver2.0.jar) installation path.

For example, in C shell, use the following command:

```
set CLASSPATH = <solidDB installation directory>/jdbc/SolidDriver2.0.jar . $CLASSPATH
```

If you are using an UNIX shell other than C shell, modify this command to make it appropriate for your shell.

Related concepts:

3.4, "License certificate," on page 43
The solid.lic license file is distributed as a separate download image called the License Certificate.

4.3.2, "Installing solidDB ODBC Driver," on page 53
The solidDB installation program installs two ODBC Drivers: one for Unicode and one for ASCII. The Unicode version is a superset of the ASCII version; you can use it with either Unicode or ASCII character sets. On Windows environments, you can also use the solidDB installation program to install only the ODBC driver.

4.3.1, "Installing solidDB JDBC Driver"
The solidDB JDBC Driver (SolidDriver2.0.jar) is installed during solidDB server installation. Depending on your environment, you might need to set various configuration settings before you use the solidDB JDBC Driver.

4.3 Installing JDBC and ODBC drivers

The solidDB JDBC Driver and solidDB ODBC Driver are included in the solidDB server installation package. Both drivers are installed by default during solidDB installation.

4.3.1 Installing solidDB JDBC Driver

The solidDB JDBC Driver (SolidDriver2.0.jar) is installed during solidDB server installation. Depending on your environment, you might need to set various configuration settings before you use the solidDB JDBC Driver.

Default installation directory

The solidDB JDBC Driver is installed during the solidDB server installation into the jdbc directory.

If your application is on a different computer than the solidDB server, you must copy the JDBC driver file to the computer where the application is located.

Tip:

- The jdbc directory contains also the solidDB Data Store Helper Class (SolidDataStoreHelper.jar) for use with WebSphere.
- The samples/jdbc directory in the solidDB installation directory contains Java code samples that use the solidDB JDBC Driver. Instructions for running the sample are available in the readme.txt file, which is available in the same directory.

Requirements for Java environment

- Ensure that you have a working Java runtime or development environment that supports JDBC API specification release 2.0.
- Check from your Java environment documentation whether it can use compressed bytecode. The SolidDriver2.0.jar contains the solidDB JDBC Driver classes in compressed bytecode format usable by most Java virtual

machines. However, some environments (such as Microsoft J++) require uncompressed bytecode. If your environment requires uncompressed bytecode, you must extract the `SolidDriver2.0.jar` file by using a tool that supports long file names.

Setting the CLASSPATH environmental variable

The CLASSPATH environment variable for your environmentmust include the solidDB JDBC Driver `.jar` file installation path.

Windows

The installation adds the default solidDB JDBC Driver installation path to the System CLASSPATH environment variable automatically.

You can check and set the System CLASSPATH environment variable through the Control Panel:

Control Panel > **System** > **Advanced** > **Environment Variables**

Linux and UNIX

Set your CLASSPATH environment variable to include the solidDB JDBC Driver (`SolidDriver2.0.jar`) installation path.

For example, in Bourne shell, use the following command:

```
export CLASSPATH=<solidDB installation directory>/jdbc/SolidDriver2.0.jar:$CLASSPATH
```

If you are using another shell than the Bourne shell, modify this command to make it appropriate for your shell.

4.3.2 Installing solidDB ODBC Driver

The solidDB installation program installs two ODBC Drivers: one for Unicode and one for ASCII. The Unicode version is a superset of the ASCII version; you can use it with either Unicode or ASCII character sets. On Windows environments, you can also use the solidDB installation program to install only the ODBC driver.

Windows

In Windows environments, the solidDB installation program installs the ODBC drivers and the following system Data Source Names (DSN) automatically. You can also add you own user DSNs.

- Windows 32-bit operating systems:
  - IBM solidDB 7.0 32-bit – ANSI
  - IBM solidDB 7.0 32-bit – Unicode
- Windows 64-bit operating systems:
  - IBM solidDB 7.0 64-bit – ANSI
  - IBM solidDB 7.0 64-bit – Unicode

Linux and UNIX

In Linux and UNIX environments, the ODBC driver library files are installed to the following directories:

- `<solidDB installation directory>/bin/`: dynamic library files
  - `sac<platform><version>.sa` or `sac<platform><version>.so` – ANSI
  - `soc<platform><version>.sa` or `soc<platform><version>.so` – Unicode
- `<solidDB installation directory>/lib/`: static library files

- `solidodbca.sa` or `solidodbca.so` – ANSI
- `solidodbcu.sa` or `solidodbcu.so` – Unicode

The file extension `.sa` or `.so` depends on the operating system.

Installing ODBC drivers without solidDB installation (Windows)

To install the ODBC drivers without installing solidDB in Windows environments:

1. Start the solidDB installation program.
2. Select **Custom** installation.
3. Select **ODBC** (clear **Server** and **Samples**).
4. Follow the displayed instructions to complete the installation.

Installing ODBC drivers without solidDB installation (Linux and UNIX)

To install the ODBC drivers without installing solidDB in Linux and UNIX environments:

1. Install solidDB using the installation program.
2. Copy the ODBC driver library file to your client node.

4.4 Installing solidDB Documentation package

If you want to access the English version solidDB documentation in PDF format on the same node where your solidDB server is installed, you can download and extract the solidDB Documentation package into the `'manuals'` directory in the solidDB installation directory.

Procedure

1. Locate the *IBM solidDB Documentation* package.
   - In physical media deliveries, the documentation package is included in the Quick Start DVD.
   - If you download your software from IBM Passport Advantage, locate the documentation package.
   - If you download your solidDB from the solidDB FTP site at ftp://ftp.software.ibm.com/software/data/soliddb/info/7.0/man/, locate the `.zip` file for the English version manuals.
2. Download and extract the documentation package into the `'manuals'` directory in your solidDB installation directory.

4.5 Installing fix packs and interim fixes

Fix packs and interim fixes are distributed as single installation program files. You can skip fix pack and interim fix levels. For example, you can upgrade from Fix Pack 3 (7.0.0.1) to Fix Pack 5 (7.0.0.5) without installing the interim fix pack (7.0.0.4).

- "Installing fixes for solidDB"
- "Installing fixes for InfoSphere CDC components" on page 55

Installing fixes for solidDB

Before you begin

Familiarize yourself with the new and changed features that are described in the solidDB 7.0 *Release Notes*, available in the **IBM solidDB 7.0 Information Center** at http://publib.boulder.ibm.com/infocenter/soliddb/v7r0/index.jsp.

1. **Make a backup copy of your database files, log files, and the `solid.ini` configuration file.**

   For instructions, see section *Performing backup and recovery* in the *IBM solidDB Administrator Guide*.
2. **Shut down solidDB.**

   For instructions, see Shutting down solidDB.
3. **Run the solidDB fix pack installer according to the instructions in Installing solidDB.**

   solidDB fix packs are delivered as single installation files. To replace your existing solidDB installation, use the same installation directory where your existing solidDB is installed.

   If you do not want to run the installer on top of your existing solidDB installation (for example, on your production environment node), use a separate directory and copy the executable program files, libraries, and drivers manually, as applicable for your setup.

Installing fixes for InfoSphere CDC components

Before you begin

To ensure that configuration information for your instances, datastores, and subscriptions is retained during the installation, complete the following steps before you start installing any InfoSphere CDC components:

1. End replication on all subscriptions.
2. Disconnect from Access Server.
3. Exit Management Console.
4. Stop all InfoSphere CDC instances in the InfoSphere CDC for solidDB Configuration Tool and InfoSphere CDC for backend data server Configuration Tool. For instructions, see Stopping InfoSphere CDC.

Installing fix packs for Access Server

Important: You must update Management Console and Access Server to the same fix pack level.

1. Uninstall the Access Server.
2. Run the Access Server fix pack installer according to the instructions in Installing Access Server.

   Note:

   - Use the same installation directory where the previous version of the Access Server was installed.
   - You cannot run two versions of the Access Server on the same node.

Installing fix packs for Management Console

Important: You must update Management Console and Access Server to the same fix pack level.

1. Uninstall the Management Console.

2. Run the Management Console fix pack installer according to the instructions in Installing Management Console.

   Note: Use the same installation directory where the previous version of the Management Console was installed.

Installing fix packs for InfoSphere CDC for solidDB

Important: To install a fix pack for InfoSphere CDC for solidDB successfully, you must already have an installation of InfoSphere CDC for solidDB.

1. Run the InfoSphere CDC for solidDB fix pack installer according to the instructions in Installing InfoSphere CDC.

   Note: Use the same installation directory where the previous version of the InfoSphere CDC for solidDB is installed; the installer prompts you to upgrade the installation.

Installing fix packs for InfoSphere CDC for *backend data server*

Important: To install a fix pack for InfoSphere CDC for *backend data server* successfully, you must already have an installation of InfoSphere CDC for *backend data server*.

1. Run the InfoSphere CDC for *backend data server* fix pack installer according to the instructions in the *IBM InfoSphere Change Data Capture, End-User Documentation* for the backend data server, included in the *IBM InfoSphere Change Data Capture Documentation* package.

   Note: Use the same installation directory where the previous version of the InfoSphere CDC for *backend data server* is installed; the installer prompts you to upgrade the installation.

Restarting replication on subscriptions after installation

After you have installed all the fix pack components:

1. Ensure that solidDB and your backend data server are running.
2. Start the InfoSphere CDC instances in the InfoSphere CDC for solidDB Configuration Tool and InfoSphere CDC for backend data server Configuration Tool.
3. Log in to Management Console.
4. Start replication on the subscriptions.

For instructions, see

- Starting InfoSphere CDC
- Management Console administration

4.6 Uninstalling solidDB

In Windows environments, uninstall solidDB through **Control Panel** > **Add or Remove Programs**. In Linux and UNIX environments, remove the installation directory.

Procedure

- **Windows:**

1. Go to **Control Panel** > **Add or Remove Programs** > **Change or Remove Programs**.
2. Select IBM solidDB, and click the **Change/Remove** button.

- **Linux and UNIX:**

1. Remove the installation directory.

5 Verifying your solidDB installation

After installing solidDB server, you can start the solidDB server and create your first database to verify that the installation was successful. You can also use the samples that are included in the package to explore the solidDB features and functionality.

After a successful installation, you can find the following directories below your solidDB installation directory:

```
<installation directory>
    bin\
    ..
    eval_kit\
        standalone\
        cdc\
    ..
    samples
    ..
```

The `standalone` directory can work as the *working directory* for starting solidDB and creating your first database in the way shown in the sections below. The `standalone` directory contains a sample `solid.ini` configuration file and an evaluation license file (`solideval.lic`). The working directory also holds your first database (`solid.db`) and the related transaction log, message, and trace files.

The `cdc` directory contains a sample `solid.ini` configuration file for use with solidDB Universal Cache. The `cdc` directory can be used as the working directory when you are setting up solidDB Universal Cache.

Both the `standalone` directory and the `cdc` directory contain a license file that you can use to evaluate solidDB for a limited time. All solidDB and solidDB Universal Cache product options and features are available for evaluation.

5.1 Working directory and `solid.ini` configuration file

A working directory is the directory that contains the files that are related to running a particular solidDB instance, including the `solid.ini` configuration file. The `solid.ini` file specifies parameters that help customize and optimize solidDB.

Working directory

Typically the working directory contains the following files:

- License file
- `solid.ini` configuration file
- Database files
- Transaction log files
- Message and trace files

In typical setups, the working directory contains the valid license file and the `solid.ini` configuration file. You can also specify a different directory for the license file and the `solid.ini` file by creating a SOLIDDIR environment variable that specifies the location.

When you start the solidDB server, it reads configuration parameters from solid.ini. You can also use the server without the configuration file, in which case the default settings (factory values) are used. Also, if solid.ini exists, but a value for a particular parameter is not set, the server uses a factory value for that parameter. The factory values might depend on the operating system you are using.

The database, transaction log, message, and trace files can be placed in other directories of your choice; in that case, the directory paths and file names must be defined in the solid.ini configuration file. For example, the **FileSpec** parameter in the [IndexFile] section of the solid.ini file specifies the directory and files name for the database file.

solid.ini configuration file for evaluation setups

The parameter settings in the sample solid.ini configuration file in the eval_kit\standalone directory serve most evaluation needs for the solidDB server.

If you want to use other components such as the Universal Cache or the High Availability feature, the configuration file must be modified accordingly.

The samples also contain solid.ini configuration files, which contain configuration settings that are needed to run the samples. When you are evaluating solidDB with the different features and functionality, the solid.ini configuration files that are provided with the corresponding samples can be used as a starting point.

Related topics

For more information about the solid.ini configuration file and how to set parameters through it to configure the solidDB server, refer to the *IBM solidDB Administrator Guide*.

5.2 Starting solidDB and creating your first database

This section describes how to start an evaluation version of solidDB, using the license file and the solid.ini configuration file in the eval_kit\standalone directory.

To start solidDB, a valid license file must be located in your working directory.

Starting solidDB in Linux and UNIX environments

1. If you have several solidDB installations on the same computer, verify that the version 7.0 installation is your default solidDB installation.
   a. To check which **solid** executable program is defined in the PATH environment variable, enter the following command at the command prompt:
      ```
      which solid
      ```
   b. To verify the version of the **solid** executable program, enter the following command at the command prompt:
      ```
      solid -x version
      ```
2. In the eval_kit/standalone directory, enter the command solid -f at the command prompt.

   The option -f forces the server to run in the foreground.

When you start solidDB, it checks whether a database already exists. If no database is found, solidDB prompts you to create a new database.

3. Type `y` to create a new database.
4. Provide the username, password, and a name for the default database catalog.

   CAUTION:
   There are no defaults for the user name and password. You must remember the user name and password to be able to access the database again.

   For example, you can use `dba` for the user name and password of evaluation databases, since it is simple and easy to remember.

   To use it:

   a. Type any name you prefer as the catalog name.
   b. Type `dba` as user name and password.
   c. Retype `dba` as the password.

Result

solidDB creates a new database called `solid.db` in the `eval_kit/standalone` directory.

The solidDB process is left running in the foreground. To make solidDB run in the background:

1. Press **CTRL+Z** to suspend the process.
2. Enter `bg` to force the process to the background.

Tip: Alternatively, you can start solidDB and create a new database by specifying the database and login data as command-line options.

For example:

```
solid -Udba -Pdba -Cdba
```

This starts solidDB in the background and creates a new database with the following login data:

- Username: dba
- Password: dba
- Catalog name: dba

Starting solidDB in Windows environments

Click the icon labeled **Start IBM solidDB server** through the **Start** > **Programs** > **IBM solidDB** menu path.

Result

When you start solidDB, it checks whether a database already exists. If no database is found, solidDB creates a new database called `solid.db` in the `eval_kit\standalone` directory automatically, with the following login data:

- Username: dba
- Password: dba
- Catalog name: dba

Tip: If you would like the **Start IBM solidDB server** icon to start solidDB and create a database in a different directory or with different login data and catalog name, you can modify the installation default settings by right-clicking the **Start IBM solidDB server** icon and editing the shortcut **Properties**. For more details, see section *Modifying Windows shortcuts* in the *IBM solidDB Administrator Guide*.

5.3 Connecting to solidDB for the first time

After you have created your first database and your solidDB process is running, you can connect to it from any client application by using the communication ports that the solidDB process is listening to. This can be done, for example, by using the command-line tool solidDB SQL Editor (**solsql**).

The communication port is defined with the **Listen** parameter in the [Com] section of the solid.ini file.

For the evaluation version of solidDB in the eval_kit\standalone directory, the following communication ports are defined:

```
[Com]
Listen=tcpip 2315, tcpip 1315, tcpip 1964
```

5.3.1 Connecting to a database with the solidDB SQL Editor

To connect to a database with solidDB SQL Editor (**solsql**), enter the solsql start command at your operating system command prompt, specifying the network name and credentials for the solidDB server you want to connect to. On Windows operating systems, you can also use the **Start** > **Programs** > **IBM solidDB** > **solidDB SQL Editor** menu path to start the editor.

Procedure

To start the solidDB SQL Editor:

- On Windows operating systems, click the icon labeled **solsql SQL Editor** through the **Start** > **Programs** > **IBM solidDB** menu path.

  This starts the solidDB SQL Editor, connecting to solidDB server using the communication port "tcp 2315" and username "dba" and password "dba".

  Tip: If you would like the **solsql SQL Editor** icon to start solsql with different login data or communication port, you can modify the installation default settings by right-clicking the **solsql SQL Editor** icon and editing the shortcut **Properties**. For more details, see section *Modifying Windows shortcuts* in the *IBM solidDB Administrator Guide*.
- In all other environments, enter the SQL editor start command at your operating system prompt. The command syntax is as follows:

  ```
  solsql "networkname" [userid [password]]
  ```

  For example:

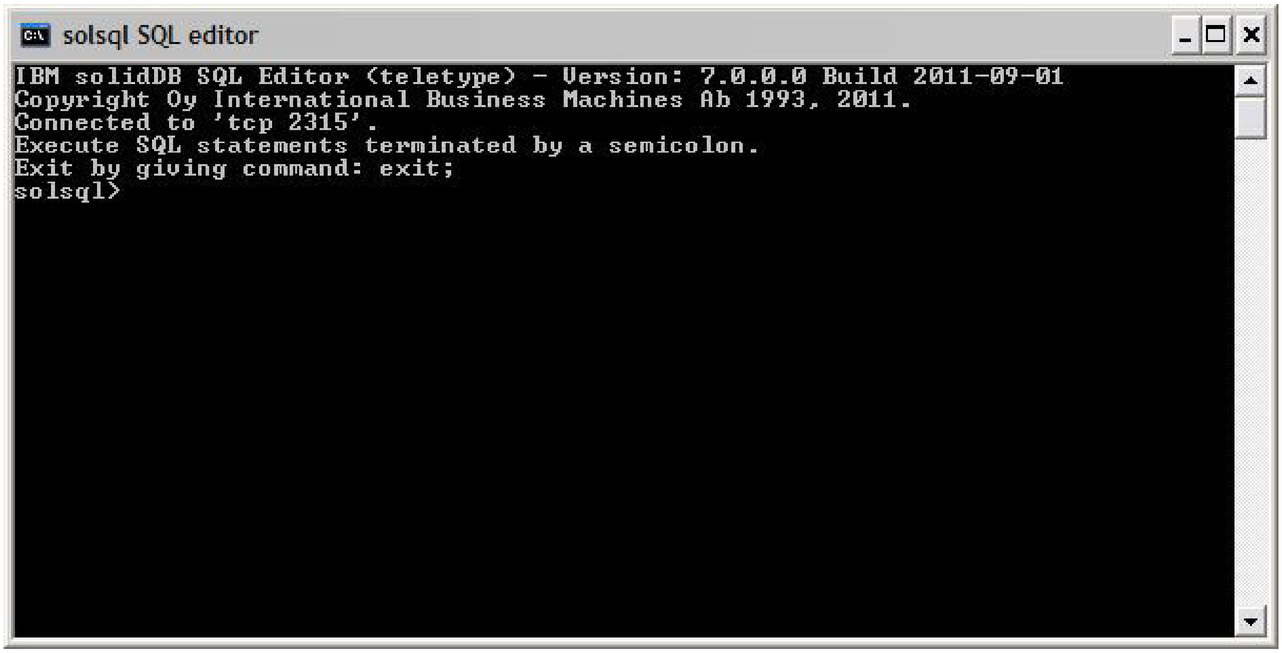
  ```
   solsql "tcp 2315" dba dba
  ```

  Tip: If you do not give your user name and password when you start the solidDB SQL Editor, you are prompted for them.

Results

If you entered a valid user name and password when you started the solidDB SQL Editor, you are connected to the database. The figure below shows the solidDB

SQL Editor after a successful connection to the database.



```
IBM solidDB SQL Editor (teletype) - Version: 7.0.0.0 Build 2011-09-01
Copyright Oy International Business Machines Ab 1993, 2011.
Connected to 'tcp 2315'.
Execute SQL statements terminated by a semicolon.
Exit by giving command: exit;
solsql>
```

Figure 14. solidDB SQL Editor connected

5.3.2 Viewing database and configuration status

You can use the solidDB SQL Editor (**solsql**), for example, to view the status of the database, check the parameter settings, or display the available administration commands (ADMIN COMMAND).

About this task

When you issue ADMIN COMMAND commands in **solsql**:

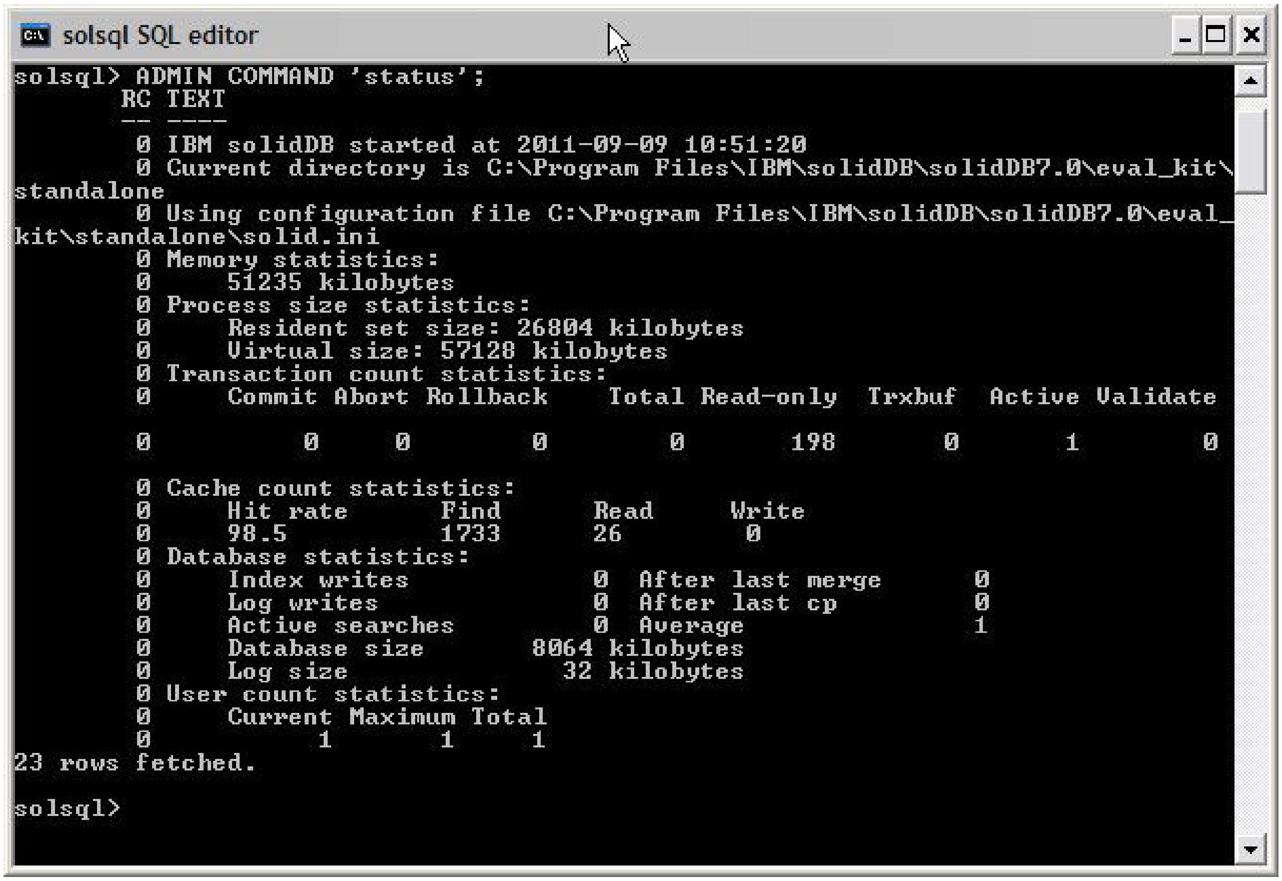
- use a semicolon at the end of the command, and
- use single quotation marks, not double quotation marks when specifying the command name.

Procedure

1. To view the status of your database, issue the following command in the solidDB SQL Editor:

   ```
   ADMIN COMMAND 'status';
   ```

   An example output from the solidDB SQL Editor is shown below:



```
solsql> ADMIN COMMAND 'status';
       RC TEXT
       -- ----
        0 IBM solidDB started at 2011-09-09 10:51:20
        0 Current directory is C:\Program Files\IBM\solidDB\solidDB7.0\eval_kit\
standalone
        0 Using configuration file C:\Program Files\IBM\solidDB\solidDB7.0\eval_
kit\standalone\solid.ini
        0 Memory statistics:
        0     51235 kilobytes
        0 Process size statistics:
        0     Resident set size: 26804 kilobytes
        0     Virtual size: 57128 kilobytes
        0 Transaction count statistics:
        0     Commit Abort Rollback    Total Read-only  Trxbuf  Active Validate

        0          0     0        0        0       198       0       1        0

        0 Cache count statistics:
        0     Hit rate      Find      Read     Write
        0     98.5          1733      26        0
        0 Database statistics:
        0     Index writes            0  After last merge      0
        0     Log writes              0  After last cp         0
        0     Active searches         0  Average               1
        0     Database size       8064 kilobytes
        0     Log size              32 kilobytes
        0 User count statistics:
        0     Current Maximum Total
        0           1       1     1
23 rows fetched.

solsql>
```

Figure 15. Example output from solidDB SQL Editor (solsql)

2. To view all parameter settings, issue the following command:

   ```
   ADMIN COMMAND 'par';
   ```

   Tip: You can view section-specific parameter settings with the solsql command:

   ```
   ADMIN COMMAND 'par section_name';
   ```

   For example:

   ```
   ADMIN COMMAND 'par IndexFile;
   ```

3. To view the available ADMIN COMMANDs, issue the following command:

   ```
   ADMIN COMMAND 'help';
   ```

5.4 Issuing SQL statements with solidDB SQL Editor

You can issue SQL statements in the solidDB SQL Editor. This section shows an example of creating, viewing, and deleting a test table.

Procedure

1. Create a table by issuing the following commands in **solsql**:

   ```
   CREATE TABLE TESTTABLE (VALUE INTEGER, NAME VARCHAR);
   COMMIT WORK;

   INSERT INTO TESTTABLE (VALUE, NAME) VALUES (31, 'DUFFY DUCK');
   ```

```
COMMIT WORK;

SELECT VALUE, NAME FROM TESTTABLE;
COMMIT WORK;
```

2. View the table with the following command:

```
SELECT VALUE, NAME FROM TESTTABLE;

    VALUE NAME
    ----- ----
       31 DUFFY DUCK
1 ROWS FETCHED.

COMMIT WORK;
```

3. Delete the table with the following command:

```
DROP TABLE TESTTABLE;
COMMIT WORK;
```

Tip: To ensure that every database operation gets committed, remember to issue the COMMIT WORK statement.

5.5 Stopping and restarting the database

5.5.1 Stopping the database with solidDB SQL Editor

To stop the database, you must prevent users from connecting to the database, disconnect all connected users, and then shut down solidDB.

About this task

You can stop the database by using the solidDB SQL Editor (**solsql**). You must include a semicolon at the end of the commands. Also, you must use single quotation marks, not double quotation marks.

Procedure

1. Prevent additional users from connecting to the database:

```
ADMIN COMMAND 'close';
```

2. Throw out all connected users (except the one who issued the command):

```
ADMIN COMMAND 'throwout all';
```

3. Shut down solidDB:

```
ADMIN COMMAND 'shutdown';
```

4. Exit from the **solsql** tool:

```
EXIT;
```

Note: The first three commands can be replaced with this one:`admin command 'shutdown force';`

Results

The solidDB database is shut down with all previously connected users disconnected.

Note:

When you shut down the server, it breaks the connection to **solsql**, and your **solsql** might show an error message such as:

14519: `The user was thrown out from the server; connection lost.`

5.5.2 Restarting the database (example)

You can restart solidDB by issuing commands or by using the **Start** > **Programs** menu on Windows operating systems.

About this task

This section provides instructions for restarting an evaluation database in the `eval_kit\standalone` directory.

Procedure

Choose one of the following methods to restart the database:

- In Windows, click the icon labeled **Start IBM solidDB server** through the **Start** > **Programs** > **IBM solidDB** menu path.
- In Linux, UNIX, or Windows command prompt, move to the solidDB installation root directory and enter the following command:

  ```
  bin\solid -c .\eval_kit\standalone
  ```

  The command-line option `-c <directory_path>` defines the working directory.
- If you have the solidDB `bin` directory in your path, go to the `'eval_kit\standalone'` directory to make it your current working directory, and start solidDB by issuing the command `solid`.

6 Using solidDB with graphical SQL clients

solidDB does not include a graphical user interface (GUI). However, graphical SQL clients such as IBM Data Studio Administrator, Eclipse SQL Explorer, or SQuirrel SQL Client can be used with solidDB to browse database objects, view and modify table data, and issue SQL commands.

- IBM Data Studio Administrator and Eclipse SQL Explorer clients are ideal if your development environment already uses the Data Studio or Eclipse frameworks.
- SQuirrel SQL Client is a stand-alone program that requires no other components than the client and solidDB to work.

All the above mentioned clients use the JDBC interface to connect to solidDB.

IBM Data Studio Administrator

IBM Data Studio Administrator is a powerful and flexible tool that helps you manage your database objects and simplifies the process of identifying, analyzing, and implementing database schema changes.

You can connect Data Studio Administrator to a solidDB database using the Data Studio Administrator's Generic JDBC database manager options.

For more details on setting up Data Studio Administrator with solidDB, see the following articles on ibm.com:

- *Configuring Data Studio Administrator V2.1 for use with solidDB*: http://www.ibm.com/support/docview.wss?rs=3457&uid=swg21411299
- *Configuring Data Studio Administrator V1.2 for use with solidDB*: http://www.ibm.com/support/docview.wss?rs=3457&uid=swg21411276

Eclipse SQL Explorer

Eclipse SQL Explorer is a thin SQL client that you can use to query and browse any JDBC-compliant database, including solidDB.

The Eclipse SQL Explorer can be added as a plugin to your existing Eclipse environment, or you can run it as a stand-alone client.

For more details on setting up SQL Explorer with solidDB, see the following article on ibm.com:

- *Configuring Eclipse SQL Explorer for use with solidDB*: http://www.ibm.com/support/docview.wss?rs=3457&uid=swg21411273

SQuirrel SQL Client

SQuirrel SQL Client is an open source Java SQL Client program for JDBC-compliant databases. It is a stand-alone client that can be configured to connect to your solidDB database via the solidDB JDBC driver.

For more details on setting up SQuirrel SQL Client with solidDB, see the following article on ibm.com:

- *Configuring Squirrel SQL for use with solidDB*: http://www.ibm.com/support/docview.wss?rs=3457&uid=swg21411262

7 Running samples

You can find sample programs and scripts in the `'samples'` directory in the solidDB installation directory. Each sample directory includes also a `readme.txt` file that provides instructions for how to use the samples.

Before you begin

To run samples:

- The sample directory must contain a valid license file. By default, each sample directory contains a copy of the evaluation license (`solideval.lic`).

  If you want to use the full product license (`solid.lic`) with the samples:

  1. Copy the license file to the root of the solidDB installation directory.
  2. Execute the script called `copy_licenses` (`copy_licenses.bat` in Windows). This script copies the license file from the installation root directory to all appropriate sample directories.

Procedure

Run the sample according to instructions provided in the `readme.txt` file, which is located in the sample directory.

Note: Some sample scripts remove and re-create database files residing in sample subdirectories. Only the databases in the `eval_kit/standalone` and `eval_kit/cdc` directories are always left intact.

8 Upgrading solidDB to a new release level

solidDB supports upgrades from at least the two previous release levels. Upgrades to solidDB 7.0 are supported from the 6.5 and 6.3 release levels; conversion of the database files is however needed. If you need to upgrade from older versions, contact IBM Software Support for assistance.

Before you begin

Familiarize yourself with the new and changed features that are described in the solidDB 7.0 *Release Notes*, available in the **IBM solidDB 7.0 Information Center** at http://publib.boulder.ibm.com/infocenter/soliddb/v7r0/index.jsp.

About this task

The steps below describe the procedure for upgrading a single solidDB server to a new release level. If you are using solidDB in High Availability mode, see section *Upgrading HotStandby servers* in the *IBM solidDB High Availability User Guide*.

The previous release level solidDB JDBC and ODBC drivers are compatible with the 7.0 release level solidDB server. However, the 7.0 versions of the drivers cannot be used with earlier versions of the server. Thus, the server must be upgraded before the drivers.

You must upgrade the drivers only if you plan to take into use the new features that are introduced at release level 7.0.

Note: If you want to convert a partial-Unicode database to support the full Unicode mode, see section *Converting partial-Unicode databases to Unicode* in the *IBM solidDB Programmer Guide*.

Procedure

1. **Make a backup copy of your database files and `solid.ini` configuration files.**

   For instructions, see section *Performing backup and recovery* in the *IBM solidDB Administrator Guide*.
2. **Shut down solidDB.**
3. **Install the new version of solidDB.**

   solidDB is delivered as a single installation file. If you do not want to run the installer on your production environment node, install solidDB on a separate node and copy the executables, libraries, and drivers manually to your production node, as applicable for your setup.

   For a description of the solidDB server package contents, see 3.1, “solidDB server package,” on page 33.

   For instructions on how to run the solidDB installer, see 4.2, “Installing solidDB,” on page 48.
4. **Update the license file.**

   Copy the new license file (`solid.lic`) from the License Certificate image to the solidDB working directory.

Note: If your 6.5 or 6.3 installation used the `soliduc.lic` license file, you must remove the `soliduc.lic` file and replace it with the `solid.lic` license file provided in the IBM solidDB 7.0 License Certificate image.

5. **Check the changes in parameter factory values and update the `solid.ini` configuration files if necessary.**

   Changes in the parameter values are described in the *Release Notes*.

6. **Check and update the solidDB JDBC and ODBC driver-related settings as applicable to your environment.**

   - solidDB JDBC Driver

     If you installed the new solidDB JDBC Driver (`SolidDriver2.0.jar`) into a different location than the previous one, add the new driver location to the CLASSPATH System environmental variable. By default, the solidDB JDBC driver is installed to the '`jdbc`' directory in the solidDB installation directory.

   - solidDB ODBC Driver
     - In Windows and Linux operating systems, define the new ODBC data sources. To use the new driver, you must either modify the existing data sources or create new ones.
     - If the application links to the solidDB ODBC driver directly, you must recompile your application.

7. **Convert the existing database files by starting solidDB with the `-x autoconvert` or `-x convert` command-line option.**

   In the solidDB working directory, issue the following command:

   ```
   solid -x autoconvert
   ```

   or

   ```
   solid -x convert
   ```

   The `-x autoconvert` option converts the database and starts solidDB.

   The `-x convert` option converts the database and exits solidDB.

   Note: After the conversion, you cannot use the database with an older server version. To downgrade a database, you must export and import the data using the solidDB tools.

Related concepts:

1.3.3, “System tools and utilities,” on page 17
The solidDB server package includes console tools for data management and administration, and command-line utilities for data export and import.

Part 3. Appendixes

Index

Notices

No portion of this product may be used in any way except as expressly authorized in writing by IBM.

This product is protected by U.S. patents 6144941, 7136912, 6970876, 7139775, 6978396, 7266702, 7406489, 7502796, and 7587429.

This product is assigned the U.S. Export Control Classification Number ECCN=5D992b.

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
1623-14, Shimotsuruma, Yamato-shi
Kanagawa 242-8502 Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Canada Limited
Office of the Lab Director
8200 Warden Avenue
Markham, Ontario
L6G 1C7
CANADA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs.

© Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, ibm.com, Solid, solidDB, InfoSphere, DB2, Informix, and WebSphere are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft and Windows are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and service names might be trademarks of IBM or other companies.

SC27-3838-05