

IBM solidDB
IBM solidDB Universal Cache
Version 6.3

*IBM solidDB Universal Cache User
Guide*



Note

Before using this information and the product it supports, read the information in "Notices" on page 87.

First edition, third revision

This edition applies to version 6, release 3 of IBM solidDB (product number 5724-V17) and IBM solidDB Universal Cache (product number 5724-W91) and to all subsequent releases and modifications until otherwise indicated in new editions.

© Oy International Business Machines Ab 1993, 2011

Contents

Figures	v	About this section	31
Tables	vii	About InfoSphere CDC for IBM solidDB.	31
Summary of changes.	ix	System requirements	31
About this manual	xi	Required database, user accounts, and schemas	32
Typographic conventions	xi	Single byte and multibyte character support	32
Syntax notation conventions.	xii	Installing InfoSphere CDC	33
1 Overview of IBM solidDB Universal Cache	1	Installing InfoSphere CDC using an interactive installation	33
Cache	1	Installing InfoSphere CDC using a silent installation	34
Architectural overview of solidDB Universal Cache	2	Configuring InfoSphere CDC (Windows)	35
Universal Cache configurations	3	Configuring InfoSphere CDC instances (Windows).	35
Features	8	Configuring InfoSphere CDC (UNIX and Linux)	38
Restrictions	8	Configuring InfoSphere CDC instances (UNIX and Linux)	38
Database restrictions.	8	Starting and stopping InfoSphere CDC	40
InfoSphere CDC restrictions	10	Starting InfoSphere CDC	40
Security and authentication	10	Stopping InfoSphere CDC	41
2 Installing solidDB Universal Cache	13	Enabling SQL statements in Management Console.	42
3 Configuring solidDB Universal Cache	15	Datatypes supported by InfoSphere CDC	43
Configuring solidDB and back-end data servers	15	Supported data types	43
Configuring solidDB	15	Supported mappings	43
Configuring back-end data server	16	InfoSphere CDC metadata tables	44
Creating InfoSphere CDC instances	16	Commands for InfoSphere CDC	45
Setting up replication subscriptions	17	Using the InfoSphere CDC commands	45
4 Performance tuning and monitoring	21	Setting the TSINSTANCE environment variable	45
Factors impacting solidDB Universal Cache performance	21	Controlling replication commands	46
Monitoring performance	22	Database transaction log commands	48
5 solidDB Universal Cache failure scenarios	25	Managing tables for replication commands	52
Standalone solidDB server fails.	25	Monitoring replication commands	58
InfoSphere CDC instance fails	25	Exporting and importing configuration commands.	61
solidDB server in HA mode (HotStandby) fails	26	Other commands	62
Communication link between primary solidDB server and InfoSphere CDC for solidDB instance fails	26	User exits for InfoSphere CDC	66
Back-end server or back-end node fails	27	Stored procedure user exits for table and row level operations	66
Back-end primary server fails	27	Defining a stored procedure user exit.	67
6 Troubleshooting	29	Stored procedure user exit database connections	67
7 InfoSphere CDC for solidDB (End-User Documentation)	31	Retrieving data with a stored procedure user exit	67
		Example of a stored procedure user exit.	73
		Sample user exits for InfoSphere CDC	74
		Conflict resolution audit table	76
		Appendix to Management Console Administration Guide	79
		Configuring user exits.	79
		System parameters for Transformation Server for solidDB.	80
		Notices	87

Figures

1.	solidDB Universal Cache architecture	2	4.	solidDB Universal Cache deployment with solidDB High Availability	7
2.	solidDB Universal Cache default deployment	5	5.	Example: solidDB Universal Cache setup with three partitioning models	22
3.	solidDB Universal Cache deployment with multiple solidDB servers	6			

Tables

1. Typographic conventions	xi	4. Hardware requirements	31
2. Syntax notation conventions	xii	5. Port requirements	32
3. Documentation for solidDB Universal Cache	13		

Summary of changes

Changes for revision 03

- Editorial corrections.

Changes for revision 02

- Section “Creating InfoSphere CDC instances” on page 16 updated: starting from 6.3.1 Fix Pack 3, the authorization code for the InfoSphere™ CDC for back-end data server is no longer needed.
- The installation of solidDB® JDBC Driver (SolidDriver2.0.jar) for InfoSphere CDC for solidDB has changed: the solidDB JDBC Driver is now included in the InfoSphere CDC for solidDB package. The InfoSphere CDC for solidDB installer installs the SolidDriver2.0.jar into the lib directory.
By default, InfoSphere CDC for solidDB uses the SolidDriver2.0.jar in the lib directory.
- solidDB catalog related restrictions clarified in section “InfoSphere CDC restrictions” on page 10: subscriptions can only include tables that are included in the solidDB default database catalog (System catalog).
- Replication of single byte and multibyte character sets clarified in section “Single byte and multibyte character support” on page 32.

Changes for revision 01

- New section added: 4, “Performance tuning and monitoring,” on page 21.
- Workarounds for using foreign keys with the Refresh feature updated in section “InfoSphere CDC restrictions” on page 10.
- Configuration parameter for setting the size of the in-memory buffer for log records corrected to **LogReader.MaxSpace** in section “Features” on page 8.
- Installation instructions updated in section 2, “Installing solidDB Universal Cache,” on page 13:
If you install InfoSphere CDC for solidDB on a different node than solidDB, you must copy the solidDB JDBC driver (SolidDriver2.0.jar) to the node where InfoSphere CDC for solidDB is installed.
- solidDB JDBC Driver related information added in section 6, “Troubleshooting,” on page 29.

About this manual

IBM® solidDB® Universal Cache is a solution for speeding up traditional disk-based SQL data servers by way of one or more solidDB in-memory database instances caching the data traffic between the applications and the data servers. IBM® InfoSphere Change Data Capture technology is used to implement the data replication between the solidDB and data server instances.

This guide provides an overview of the solidDB Universal Cache as well as instructions for installing and configuring the solidDB Universal Cache. Guidelines for handling failure and troubleshooting scenarios are also included. The last section of this guide contains detailed instructions for how to install and configure the InfoSphere CDC for solidDB. This section is needed when configuring the solidDB Universal Cache; it provides comparable information to the *InfoSphere Change Data Capture, End-User Documentation* user manual for your back-end data server.

This manual assumes that the reader has general DBMS knowledge, and familiarity with SQL and solidDB.

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.
<i>argument</i>	Words emphasized like this indicate information that the user or the application must provide.
<i>Administrator Guide</i>	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.

Table 1. Typographic conventions (continued)

Format	Used for
File path presentation	Unless otherwise indicated, file paths are presented in the UNIX [®] format. The slash (/) character represents the installation root directory.
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 <i>table_name</i>	Syntax descriptions are on this font. Replaceable sections are on <i>this</i> 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.

1 Overview of IBM solidDB Universal Cache

IBM solidDB Universal Cache is a solution for speeding up traditional disk-based databases. It incorporates a high-speed solidDB front-end in-memory database that allows the processing of the application load by a fraction of the response time it takes to be executed in a back-end database. This improves performance, speed and latency.

In addition to the solidDB front-end database, the solution utilizes IBM InfoSphere Change Data Capture (referred to as InfoSphere CDC) technology for data replication. Both the front-end and back-end databases are self-contained database management systems (DBMSs).

Note: solidDB Universal Cache should not be confused with the internal cache (buffer pool) of solidDB. The internal cache is used to facilitate efficient I/O operations in the solidDB database server.

Architectural overview of solidDB Universal Cache

The following diagram illustrates the architecture and key components of a typical configuration of the solidDB Universal Cache.

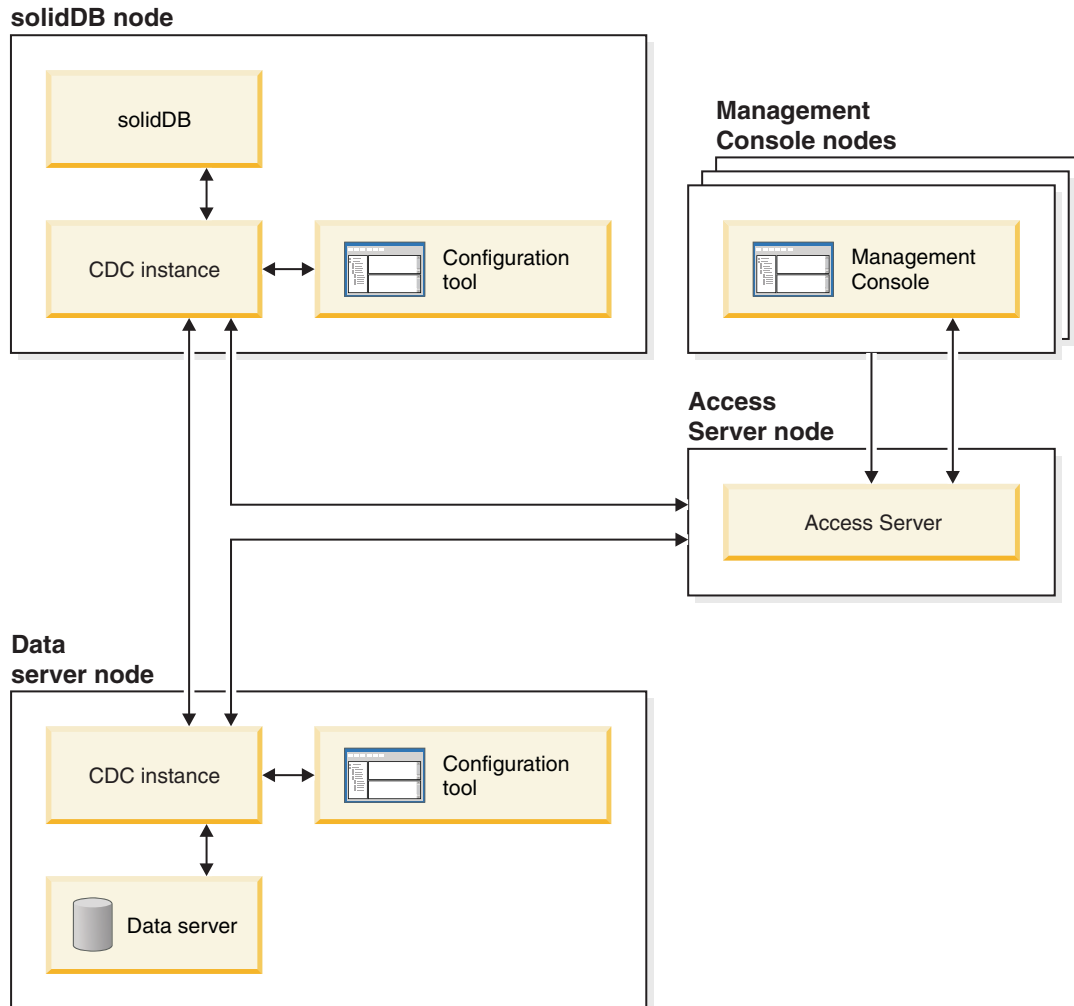


Figure 1. solidDB Universal Cache architecture

The roles and functions of the components are described below.

solidDB — the front-end database which you replicate data to and from. Depending on the replication model, solidDB can be the source or target database, or both.

Data server — the back-end database which you replicate data to and from. Depending on the replication mode, the back-end database can be the source or target database, or both.

InfoSphere CDC instance — a run-time instance of the InfoSphere CDC engine for a given DBMS. To set up InfoSphere CDC instances, you have to have the corresponding InfoSphere CDC agent software (like InfoSphere CDC for DB2®, or InfoSphere CDC for solidDB) installed on the same node where the DBMS in

question is running. Exceptionally for solidDB, the InfoSphere CDC for solidDB agent can be installed and set up on any node connected to the node running solidDB.

Configuration tool — a GUI or console based tool used to configure and create InfoSphere CDC instances. While configuring the instance, you supply the port number for communication with the rest of the InfoSphere CDC system as well as login information to connect to the database.

Access Server — a process (running typically as a service or daemon) allowing the Management Console users to access the InfoSphere CDC instances and configure them. Different users may have access to different instances. During the Access Server installation you are requested to provide a port number and Administrator login information, to be used by the Management Console.

Management Console — an interactive application with a GUI that you can use to configure and monitor replication. It allows you to manage replication on various servers, specify replication parameters, and initiate refresh and mirroring operations from a client workstation.

You start using the Management Console by creating datastores in the Access Manager perspective. A *datastore* is a logical entity of a database and the related InfoSphere CDC instance. Using the datastores defined, you can set up *subscriptions* that embody data replication from one datastore to another.

There are three types of datastores: source, target, and dual. Dual datastore can participate in subscriptions as both a source and target. Typically, dual datastores are used with solidDB Universal Cache, and symmetric mapping pairs are defined for two-way replication. If a replica is meant to be read-only, in the front-end, a single upload subscription is defined.

After you have finished setting up replication, the Management Console can be closed on the client workstation without affecting active data replication activities between source and target servers. Management Console also includes an event log and a monitor. The event log allows you to examine generated InfoSphere CDC event messages. The monitor provides the necessary support to continuously monitor replication operations and latency. Diagrams depicting components of your replication configuration are constructed through direct manipulation of graphical objects. The monitor in Management Console is intended for time-critical working environments that require continuous analysis of data movement.

Universal Cache configurations

solidDB Universal Cache supports a variety of configurations. In typical configurations, the data server and the corresponding InfoSphere CDC engine (instance) are located on the same node while the Management Console will run on a separate node. In configurations using solidDB High Availability (HotStandby), the CDC instance must run on a different node than the solidDB server.

The following sections describe common deployment topologies, showing different product components executed on different machines. Other implementations are also possible.

General principles

The deployment of the InfoSphere CDC instance depends on the database product used:

- Typically, the InfoSphere CDC instance is created on each node participating in InfoSphere CDC replication.
- The solidDB server and the InfoSphere CDC for solidDB instance do not need to be located on the same node.

This is because InfoSphere CDC for solidDB can both read and insert data into a solidDB database using both local and remote JDBC connection.

- In configurations using solidDB High Availability (HotStandby), the CDC instance must run on a different node than the solidDB server.
- The InfoSphere CDC instance for the back-end server can usually be located only on the same node as the data server itself; unlike solidDB, all databases do not provide a Log Reader API that is functional through a remote JDBC connection.

There can be several solidDB servers in the Universal Cache deployment, but only one back-end data server.

solidDB Universal Cache default deployment

The simplest setup includes the back-end data server and a single solidDB database. These are expected to run on their respective dedicated hardware. InfoSphere CDC instances associated with each database will run on the same node as the database. In production environments the InfoSphere CDC Management Console will run on a separate administrative node to limit its impact on performance of either database server.

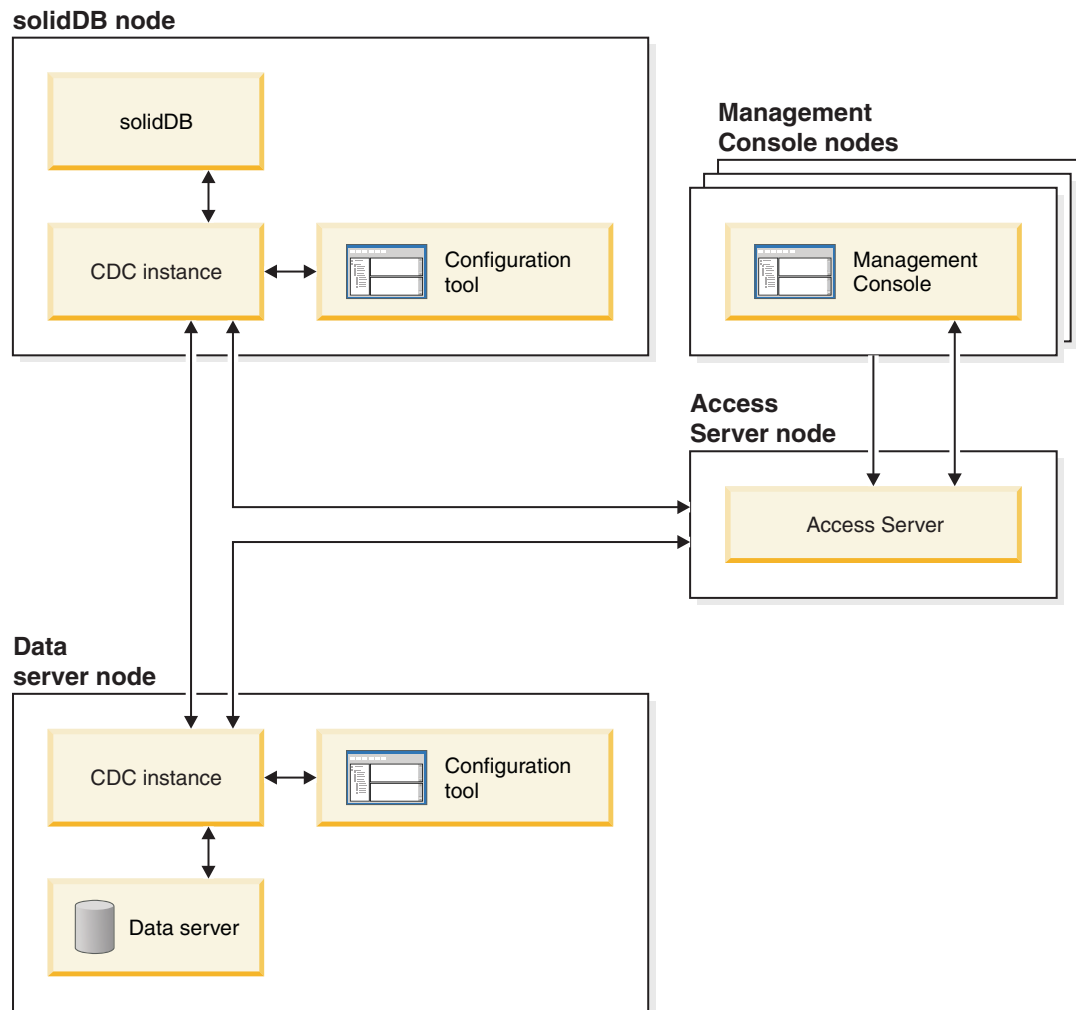


Figure 2. solidDB Universal Cache default deployment

solidDB Universal Cache deployment with multiple solidDB servers

Multiple solidDB servers can be used, for example, for partitioning back-end data over several solidDB servers in the front-end. However, in such a deployment, each solidDB server is autonomous and will process the application requests without accessing data in any of the other solidDB servers.

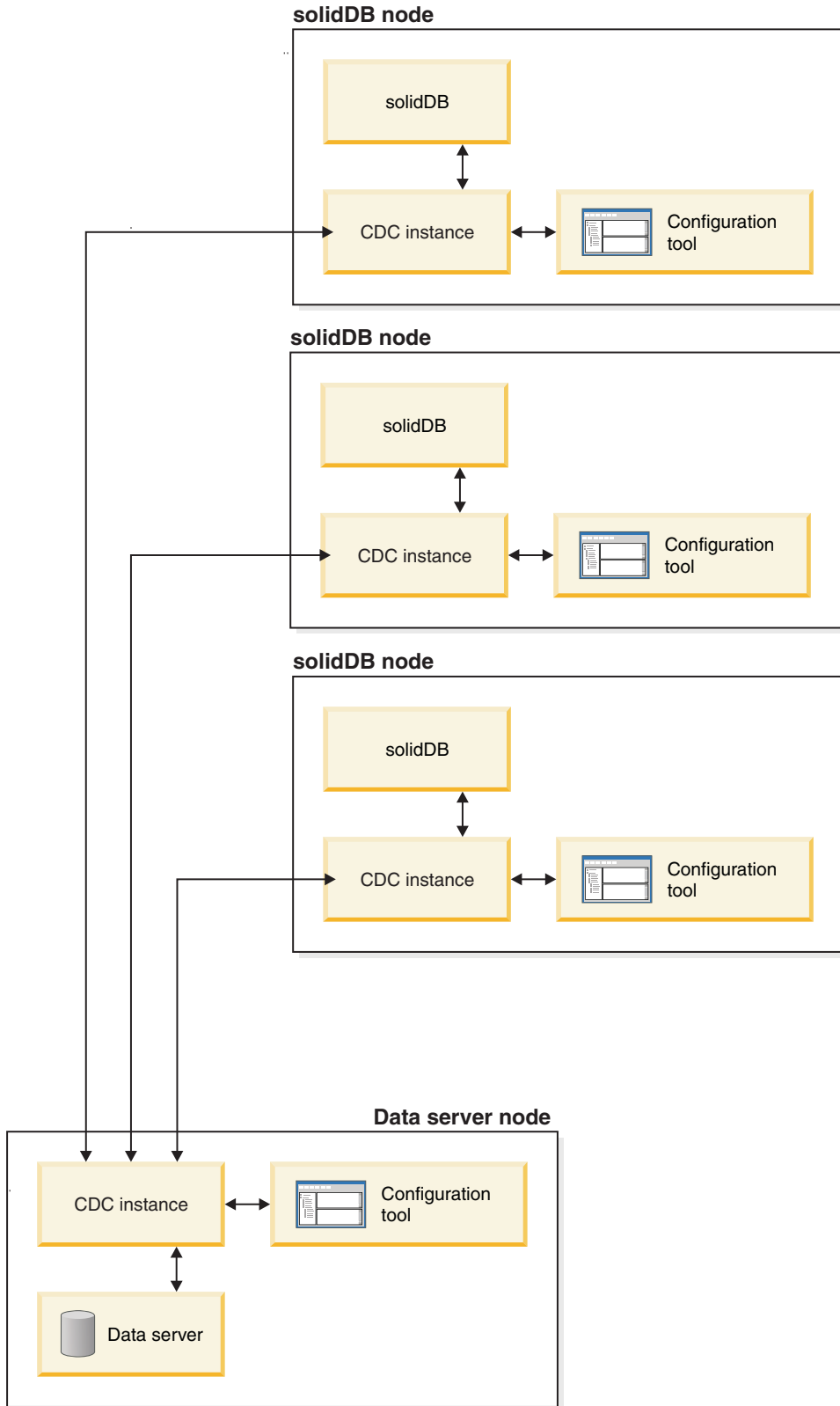


Figure 3. solidDB Universal Cache deployment with multiple solidDB servers

solidDB Universal Cache deployment with solidDB High Availability

If the front-end application requires uninterrupted access to data, solidDB HotStandby technology can be used to provide high availability. All HotStandby aspects, such as multiple durability semantics and read on standby, are applicable to solidDB Universal Cache High Availability setups. In this case all InfoSphere CDC instances will run on the back-end node, and the connection to the solidDB HotStandby pair will be established remotely. The Management Console will run on a separate node.

By using solidDB High Availability, the operation of Universal Cache is protected against single failures in the front-end tier.

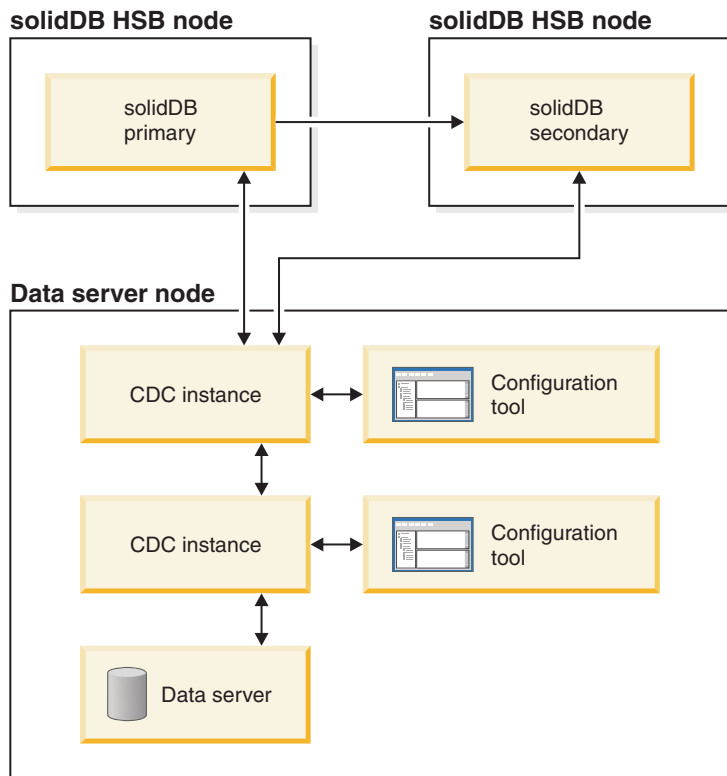


Figure 4. solidDB Universal Cache deployment with solidDB High Availability

When designing deployments with solidDB HA, remember to plan how failure scenarios are handled in your environment. See section 5, “solidDB Universal Cache failure scenarios,” on page 25 for information on how failures are handled in the solidDB Universal Cache system.

solidDB Universal Cache deployment with back-end High Availability

Applications may have additional high availability or disaster recovery requirements on the back-end data server as well. Different data servers implement such features using different technologies. The back-end data server is still a single logical entity, but data and running processes may be distributed across multiple nodes.

In these cases the InfoSphere CDC instances will run on the primary back-end data server node. However, in case of the back-end failure, these InfoSphere CDC instances will have to be restarted on a different machine as a part of the complete back-end data server failover. The subscriptions will also have to be re-configured to reconnect the relevant instances.

Features

The following solidDB features are available for configurations deploying InfoSphere CDC technology:

- **Linked library access**

solidDB applications can be built with the linked library access (ODBC and JDBC Accelerator libraries) to utilize fully the in-memory database performance and responsiveness. For more information, see the *IBM solidDB Linked Library Access User Guide*.

- **solidDB High Availability support**

Failovers in the solidDB server are supported. For more information, see the *IBM solidDB High Availability User Guide*.

- **Throttling**

If the replication cannot accommodate a sustained load in the solidDB server, the processing is throttled down (slowed). From the application standpoint, that results in increased response times. The replication traffic is buffered so that load bursts can be accommodated. The size of the corresponding in-memory buffer is controlled with the configuration parameter **LogReader.MaxSpace**. For more information, see the logreader configuration parameters in the *LogReader Section* of the *Server-Side Configuration Parameters* section in the *IBM solidDB Administrator Guide*.

- **Off-line operation and log overflow**

If the replication is stopped, or it has failed, the solidDB server may continue to process the load, and accumulate the data for a later transfer. The limit of the accumulated data is set with the **LogReader.MaxLogSize** configuration parameter. When the amount of accumulated data exceeds the values of the **LogReader.MaxLogSize** parameter, the log overflow occurs, resulting in a situation where replication catchup is no longer possible. In that case, a refresh on the subscription is needed. For more information, see the logreader configuration parameters in the *LogReader Section* of the *Server-Side Configuration Parameters* section in the *IBM solidDB Administrator Guide*.

- **Log reader diagnostics**

A number of run-time counters is available to monitor the operation of the solidDB capture (solidDB as a source) and apply (solidDB as a target) processes. See also the LOBS related restriction in the section "Database restrictions."

- **Data type support**

All solidDB data types are supported in the replication. For more information, see the section "Supported data types" on page 43.

Restrictions

Database restrictions

The following restrictions apply to solidDB as a source and target data server.

- **Referential integrity** (solidDB as a source and target)

For continuous mirroring, referential integrity constraints (foreign keys) are allowed both on the source and target. The mandatory requirement is that the referential integrity associations are to be confined within the subscription; no foreign keys may point to tables outside the subscription. If this rule is violated, referential integrity errors may happen on the target during the mirroring, which ends the replication subscription.

Referential integrity is not supported with automatic creation of tables or refresh. For a workaround, see the section “InfoSphere CDC restrictions” on page 10.

- **Datatype support**

- No D-table LOB support (solidDB as a source)

All the LOBs maintained in M-tables (in-memory tables), within the available size limits, called miniLOBS, are allowed. The size limit depends on the row size and the block size. With an assumption of a single LOB per row, the size limit is close to the block size. If the block size is set to 32KB, a practical miniLOB size limit is about 30KB.

The large-size (up to 2GB) LOBs (maxiLOBs) of solidDB D-tables (disk-based tables) are not supported in the source. This restriction is enforced in the following way: the attempt to write a maxiLOB into a D-table, which is a part of a logreader partition, fails, and an error is returned to the application.

- Limited LOB support (solidDB as a target)

If a LOB is written into an M-table and exceeds the miniLOB size limit, an error is returned and replication on the subscription ends.

- **TRUNCATE** (solidDB as a source)

The TRUNCATE TABLE statement is not allowed on tables which are part of a subscription. If this rule is violated, an error is returned to the application.

- **Primary key constraints** (solidDB as a source)

Primary keys are recommended but not mandatory. If no primary key is defined on a table, the execution of inserts and updates is less efficient than with primary keys. Primary key updates are restricted in the following way:

- If the primary key is defined over a single column, no multi-row updates of the primary key are allowed.
- In the case of multicolumn primary keys, multi-row updates are allowed if they affect a part of the primary key only.

If any of the above rules are violated, an error is produced and replication on the subscription (mirroring) ends.

- **Transient and temporary tables** (solidDB as a source)

If solidDB is used as a source, neither transient nor temporary tables can be part of the subscription.

- **Multiple NULLs in UNIQUE columns** (solidDB as a target)

In solidDB, there may be at most one NULL instance in a column defined as UNIQUE. An effort to propagate insertion of an additional NULL will result in the UNIQUE constraint violation and replication on the subscription (mirroring) ends.

- **Data and workload partitioning using multiple solidDB servers**

Multiple solidDB servers can be used for data and workload partitioning; the back-end data can be distributed (partitioned) over several solidDB servers in the front-end. However, each solidDB server is autonomous and will process the application requests only within the partition it holds, without accessing data in any of the other solidDB servers (partitions).

Referential integrity constraints apply also; the partition cannot contain tables which are referencing or referenced outside the partition.

InfoSphere CDC restrictions

The following features that may be available in the InfoSphere CDC components for other data servers are not supported in InfoSphere CDC for solidDB.

- **Fast load for refresh**

InfoSphere CDC for solidDB does not support the fast load for refresh feature.

- **Refresh**

The InfoSphere CDC refresh feature is restricted to the cases where the participating tables on the target side of a subscription are neither referencing nor referenced tables, in terms of referential integrity constraints.

If foreign keys are needed on the target side, there are two alternatives:

- **Alternative 1**

1. Delete (truncate) the tables in the target database.
2. Perform refresh.

- **Alternative 2**

1. If the existing target tables have to be refreshed, remove the foreign key constraints from the tables.
2. Create the target tables without foreign keys.
3. Perform refresh.
4. Add foreign key constraints.

- **Automatic creation of target tables**

If the tables meant to be mirrored are associated with referential integrity constraints, the option of creating the target tables automatically (**Create new target tables**), while defining a new subscription, cannot be used. Instead, the option **Map to existing tables** should be used. If this rule is violated, the subscription will not be created.

The restriction applies to all configurations, also involving other DBMS products.

- **Row filtering**

Row filtering (horizontal partitioning) is fully functional only if primary keys are defined on the source tables.

- **Dropping and recreating tables when solidDB is the source datastore**

If you need to drop and recreate tables in subscriptions where solidDB is the source datastore, you must reconfigure the table mappings:

1. Stop the replication on the subscription.
2. Remap the source tables.
3. Restart the replication (mirroring) on the subscription.

- **Subscriptions can only include tables that are included in the solidDB default database catalog (System catalog).**

The solidDB syntax for database object hierarchy is the following:

```
catalog_name.schema_name.database_object
```

The default catalog name (System catalog) is defined when the database is created and it cannot be changed later.

For more details on solidDB catalogs and object hierarchy, see *Managing database objects* in the *IBM solidDB SQL Guide*.

Security and authentication

The databases participating in the replication are protected by a normal user name/password mechanism.

- User names used for access to databases by the InfoSphere CDC instances have to be given all the necessary credentials with respect to accessing and modifying the data and metadata. A database administrator username is recommended.
- User names and passwords are stored by the InfoSphere CDC system in the InfoSphere CDC instances and their respective persistent repositories. The stored authentication data is scrambled, that is, encrypted using a weak encryption method.
- The traffic between the InfoSphere CDC instances and the databases is encrypted in all cases. The encryption is provided, or available, in the product-specific JDBC drivers.
- The traffic between InfoSphere CDC instances is unencrypted.

2 Installing solidDB Universal Cache

To deploy solidDB Universal Cache, you need to install solidDB, your back-end data server and the related InfoSphere CDC components. This section provides a high-level overview of the installation, with references to more detailed installation instructions.

1. Install IBM solidDB.

For instructions, see section *Installing solidDB* in the *Getting Started Guide*.

2. Install the back-end data server.

For instructions, see the documentation that came with your back-end data server.

3. Install InfoSphere CDC for IBM solidDB.

For details, see section “Installing InfoSphere CDC” on page 33.

Note: At the end of the installation, the installer enables you to start the Configuration Tool for creating a new InfoSphere CDC instance. Do not select to start the configuration tool unless you have completed the solidDB configuration steps as instructed in section “Configuring solidDB” on page 15.

Note: From InfoSphere CDC for IBM solidDB 6.3 Fix Pack 3 onwards, the solidDB JDBC driver (SolidDriver2.0.jar) is included in the InfoSphere CDC for solidDB package. The InfoSphere CDC for solidDB installer installs the SolidDriver2.0.jar into the lib directory.

By default, InfoSphere CDC for solidDB uses the SolidDriver2.0.jar in the lib directory.

4. Install InfoSphere CDC for the back-end data server.

For instructions, see the *InfoSphere Change Data Capture, End-User Documentation* that came with the InfoSphere CDC component for the back-end data server.

Note: At the end of the installation, the installer enables you to start the Configuration Tool for creating a new InfoSphere CDC instance. Do not select to start the configuration tool unless you have completed the back-end data server configuration steps as instructed in *InfoSphere Change Data Capture, End-User Documentation* for your back-end database.

5. Install InfoSphere CDC Access Server.

For instructions, see the *InfoSphere Change Data Capture Access Server and Management Console, Installation Guide* that came with the product.

6. Install InfoSphere CDC Management Console

For instructions, see the *InfoSphere Change Data Capture Access Server and Management Console, Installation Guide* that came with the product.

7. Check that you can access the documentation listed in the table below.

Table 3. Documentation for solidDB Universal Cache

Component	Documentation location	Key documents
InfoSphere CDC for solidDB	Included in the <i>IBM solidDB 6.3 Documentation</i> package	<i>IBM solidDB Universal Cache User Guide</i> , section <i>InfoSphere CDC for solidDB</i>

Table 3. Documentation for solidDB Universal Cache (continued)

Component	Documentation location	Key documents
InfoSphere CDC for the back-end data server	Included in the <i>InfoSphere Change Data Capture v6.3 Documentation</i> package	<i>InfoSphere Change Data Capture, End-User Documentation</i>
InfoSphere CDC Management Console	Included in the <i>InfoSphere Change Data Capture v6.3 Documentation</i> package	<i>InfoSphere Change Data Capture Management Console, Administration Guide</i>
InfoSphere CDC Access Server	No separate documentation, see the Management Console documentation	No separate documentation, see the Management Console documentation

3 Configuring solidDB Universal Cache

To configure solidDB Universal Cache, you need to first configure the front-end and back-end data servers for use with the Universal Cache, then create InfoSphere CDC instances corresponding to the data servers, and finally, set up replication subscriptions between them.

Configuring solidDB and back-end data servers

To be able to use the solidDB and back-end data servers with the InfoSphere CDC technology, you need to define various configuration settings so that the InfoSphere CDC instances can connect to and replicate data between your databases.

Configuring solidDB

To be able to use solidDB with the InfoSphere CDC technology you need modify configuration settings so that the InfoSphere CDC for solidDB can connect to and replicate data from your solidDB database. You must also set the CLASSPATH environment variable to identify the location of solidDB JDBC driver that provides InfoSphere CDC for solidDB with access to the solidDB database.

Procedure

1. Configure LogReader by modifying the configuration parameters in the LogReader section of the solid.ini configuration file.

- a. **Set the LogReaderEnabled configuration parameter to 'yes'.**

```
[LogReader]
LogReaderEnabled=yes
```

This is needed in order to be able to use solidDB as a source database in InfoSphere CDC replication. The factory value is 'no'.

- b. **Set the transaction log retention space size with the MaxLogSize configuration parameter .**

```
[LogReader]
MaxLogSize=<MB>
```

The factory value is 10 240 (10 GB). The specified log file retention space is always fully used, if the log reader is enabled. The log files may occupy even more space, if backups are not performed or the parameter **CheckpointDeleteLog** is not set to 'Yes'.

- c. **Set the in-memory buffer size for log records with the MaxSpace configuration parameter.**

```
[LogReader]
MaxSpace=<number of log records>
```

The factory value is 100000 log records. The size of a log record is that of the (binary) row size, plus a few bytes of additional metadata overhead.

When the buffer fills up, throughput throttling is applied; the operations are blocked until there is room in the logreader buffer.

2. **Modify other configuration parameters as necessary.**
 - **DurabilityLevel**

By default, the solidDB server durability level is set to relaxed (**DurabilityLevel=1**). This may result in a situation where most recent transactions may be lost if solidDB server fails unexpectedly.

To prevent this, set the durability level to strict, with the following setting in 'solid.ini' file:

```
[Logging]
DurabilityLevel=3
```

Note: Strict durability setting induces a performance penalty when compared to relaxed durability. Relaxed durability can be used without the risk of data loss if solidDB HA (HotStandby) configuration is applied with the 2-Safe replication protocol (default).

- **DefaultStoreIsMemory**

By default, the solidDB table storage type is set to M-table (**DefaultStoreIsMemory=yes**).

- **IsolationLevel**

By default, the solidDB isolation level is set to Read Committed (**IsolationLevel=1**).

3. **Check that the CLASSPATH environment variable has been set so that it includes the solidDB JDBC driver (SolidDriver2.0.jar) installation path.**

- In Windows environments, check that the solidDB JDBC driver has been set in the System variables, not (only) in User variables.

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

Control Panel > System > Advanced > Environment Variables

- If your InfoSphere CDC for solidDB is installed on a different node than solidDB, you must copy the solidDB JDBC driver (SolidDriver2.0.jar) to the node where InfoSphere CDC for solidDB is installed and set the CLASSPATH System variable accordingly.

The solidDB JDBC driver (SolidDriver2.0.jar) is located in the 'jdbc' directory under the solidDB installation directory.

Configuring back-end data server

To be able to use your back-end data server with the InfoSphere CDC replication technology, you may need to change its configuration settings.

Procedure

Configure your back-end data server according to the instructions in the *InfoSphere Change Data Capture, End-User Documentation* provided with the product.

Creating InfoSphere CDC instances

The InfoSphere CDC instances are created with the InfoSphere CDC Configuration Tool.

Before you begin

- Check that your solidDB and back-end databases are running.
- You have sufficient access privileges to your databases.

Procedure

1. **Create a new instance of InfoSphere CDC for solidDB.**

For detailed instructions, see section “Configuring InfoSphere CDC (Windows)” on page 35 or “Configuring InfoSphere CDC (UNIX and Linux)” on page 38.

Note: If your configuration deploys solidDB High Availability, you need to define the host address and port number for the primary and secondary solidDB servers.

2. Create a new instance of InfoSphere CDC for the back-end data server.

For detailed instructions, see section *Configuring InfoSphere CDC* in the *InfoSphere Change Data Capture, End-User Documentation* for the back-end data server.

Setting up replication subscriptions

The replication subscriptions are created in the Management Console. This section provides a high-level overview of the process with references to more detailed instructions in the *InfoSphere Change Data Capture Management Console, Administration Guide*.

Before you begin

- Check that the tables you intend to replicate exist on both the solidDB and the back-end databases. It is also possible to create tables during the replication if the tables do not contain foreign keys.
- Check that your solidDB and back-end databases are running.
- Check that your CDC instances for solidDB and back-end data servers are running.
- You have sufficient access privileges to your databases.
- You have defined your desired replication principles in accordance with your business rules.

Procedure

1. Log into Management Console / Connect to Access Server.

For detailed instructions, see section *Logging into Management Console (Connecting to Access Server)* in the *InfoSphere Change Data Capture Management Console, Administration Guide*.

Tip: To be able to work in the Access Manager perspective of the Management Console, you must be a System Administrator that has the privilege to manage datastores and user accounts. The System Administrator account was created during the installation of the Management Console.

2. Set up datastores for solidDB and the back-end database.

- a. Add a new datastore. For detailed instructions, see section *To add a new datastore* in the *InfoSphere Change Data Capture Management Console, Administration Guide*.
- b. Set connection parameters. For detailed instructions, see section *To set connection parameters on a datastore* in the *InfoSphere Change Data Capture Management Console, Administration Guide*.

3. Assign datastores to users.

You need to assign both the solidDB datastore and the back-end datastore to the same users.

- a. If necessary, add new users. To be able to add new users, you must be a System Administrator that has the privilege to manage user accounts. For detailed instructions, see section *Managing user accounts* in the *Management Console Administration Guide*.
 - b. Assign datastores to users. For detailed instructions, see section *Assigning users to datastores* in the *Management Console Administration Guide*.
 - c. To make the changes effective, disconnect and reconnect to the Access Server.
 - 1) Click File > Access Server > Disconnect.
 - 2) Click File > Access Server > Connect.
4. **Optionally, set the connection preferences for connecting to the datastores.**
You can also continue with the default settings.
For detailed instructions, see section *Setting connection preferences* in the *InfoSphere Change Data Capture Management Console, Administration Guide*.
5. **Set up datastores for replication.**
- a. Connect to the solidDB datastore.
 - b. Connect to the back-end datastore.
- For detailed instructions, see section *Setting up datastores for replication* in the *InfoSphere Change Data Capture Management Console, Administration Guide*.
6. **Optionally, set system parameters on the solidDB and back-end datastores.**
For detailed instructions, see section *Setting system parameters on source and target datastores* in the *InfoSphere Change Data Capture Management Console, Administration Guide*.
7. **Set up subscriptions.**
- a. Add a subscription where solidDB is the source and the back-end datastore is the target.
 - b. Add a subscription where the back-end datastore is the source and solidDB is the target.
- For detailed instructions, see section *Setting up subscriptions* in the *InfoSphere Change Data Capture Management Console, Administration Guide*.
8. **Map tables for replication in both subscriptions.**
For detailed instructions, see section *Mapping tables* in the *InfoSphere Change Data Capture Management Console, Administration Guide*.
- Tip:** To implement the Universal Cache, you should select Mirror (Change Data Capture) for the replication method.
9. **For each table mapping, set conflict detection and resolution in accordance with your business rules.**
For detailed instructions, see section *Setting conflict detection and resolution* in the *InfoSphere Change Data Capture Management Console, Administration Guide*.
10. **Start replication on both subscriptions.**
For detailed instructions, see section *Starting and ending replication on subscriptions* in the *InfoSphere Change Data Capture Management Console, Administration Guide*.

What to do next

Note:

If you need to drop and recreate tables in subscriptions where solidDB is the source datastore, you must reconfigure the table mappings:

1. Stop the replication on the subscription.
2. Remap the source tables.
3. Restart the replication (mirroring) on the subscription.

4 Performance tuning and monitoring

The performance of solidDB Universal Cache depends on a number of system and setup considerations. The Management Console monitoring features and the solidDB performance counters provide means for monitoring and analyzing the performance level.

Factors impacting solidDB Universal Cache performance

You can improve solidDB Universal Cache performance, for example, by optimizing your system and network setup or introducing parallelism to your configuration setup.

The guidelines given in this chapter complement the tuning information specific to solidDB (see *IBM solidDB Administrator Guide*) and your back-end data server.

Optimizing system and network setup

- The amount of memory allocated to each InfoSphere CDC instance should be at least 256 MB.

The memory allocation for the InfoSphere CDC instance is defined when creating the instance with the Configuration Tool (option **Maximum Memory Allowed**). The minimum allocation is 64 MB; the default is 512 MB for a 32 bit instance and 1024 MB for a 64 bit instance.

- Available CPUs

Processing data with InfoSphere CDC is CPU-intensive; ensure that you have enough CPU capacity available in all the nodes included in your solidDB Universal Cache setup.

- Network latency and throughput

Optimizing your network for high throughput and low latency improves solidDB Universal Cache performance.

Improving performance with parallelism

In a typical setup, the InfoSphere CDC engine sets the following limitations to the performance:

- At the source end, the performance is limited to about total of 15 000 operations per second for all the subscriptions together.
- At the target end, the performance is limited to about 5 000 operations per second per subscription.

The performance can be improved by using multiple subscriptions or multiple solidDB front-ends to partition the data and workload. This is because multiple subscriptions are processed in parallel at both the front-end and the back-end.

For example, you can create separate subscriptions for autonomous tables that are not referencing or referenced outside the subscription. You can also place such tables/subscriptions in separate solidDB front-ends.

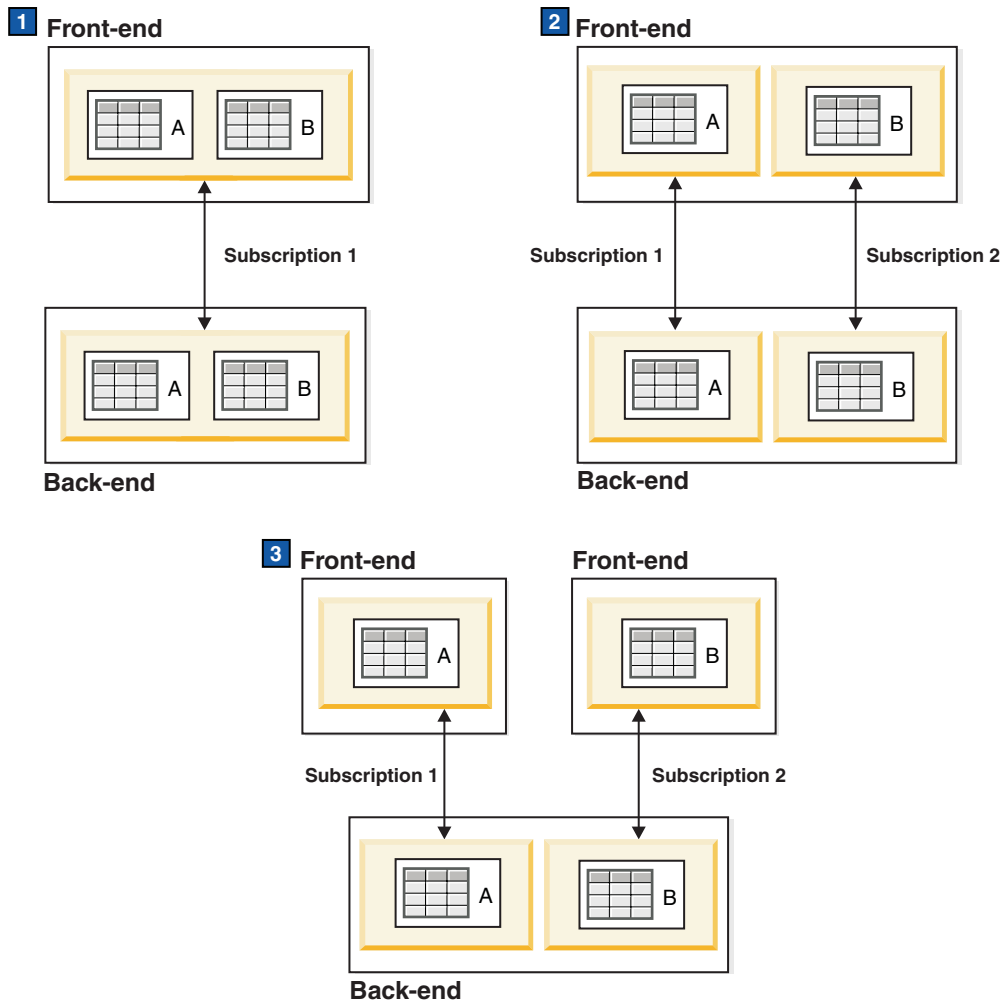


Figure 5. Example: solidDB Universal Cache setup with three partitioning models

1. Single solidDB front-end with one subscription; no partitioning
2. Single solidDB front-end with two subscriptions; partitioning with two parallel subscriptions
3. Two solidDB front-ends with one subscription each; partitioning with two front-ends

Other considerations

- The target database must be capable of bearing the load generated by the InfoSphere CDC engine.
- Any processing of data may introduce bottlenecks, for example:
 - Row level filtering
 - Data transformations and expressions
 - Codepage conversion

Monitoring performance

The Management Console **Monitoring** and **Statistics** views can be used for collecting performance statistics on the subscriptions. The solidDB performance counters provide performance data on the solidDB front-ends.

Monitoring performance in the Management Console

The Management Console can collect statistics on latency, throughput, and the number and size of the replication operations. The statistics can be viewed in Management Console or saved and exported in .csv format. You can also set latency notifications and thresholds.

For detailed instructions on how to use the monitoring and statistics in Management Console, see the section *Monitoring your subscriptions* in the *InfoSphere Change Data Capture Management Console, Administration Guide*.

Monitoring solidDB front-end performance

solidDB provides a number of performance counters specific to the use of solidDB with InfoSphere CDC:

- Counters with the variable name starting with *Logreader*

For example, *Logreader commits sent* tracks the number of commits sent to the InfoSphere CDC instance per second.

- *TS applied transactions*

The *TS applied transactions* counter tracks the number of transactions applied into solidDB by InfoSphere CDC instance when solidDB is a target datastore.

For a detailed list of the solidDB performance counters and how to use them, see section *Monitoring solidDB* in the *IBM solidDB Administrator Guide*.

5 solidDB Universal Cache failure scenarios

The following sections provide an overview of different failure scenarios and the required recovery procedures, if any.

Tip: If the recovery instructions contain manual tasks, they can often be automated by using scripts or the commands available with InfoSphere CDC.

Standalone solidDB server fails

If a standalone solidDB server fails, replication on the subscriptions ends also. To recover, proceed as follows:

Procedure

1. Restart the solidDB server manually and recover the database.
For instructions, see the section *Administering solidDB* in the *IBM solidDB Administrator Guide*.
2. Restart the InfoSphere CDC instance.
For instructions, see the section “Starting and stopping InfoSphere CDC” on page 40.
3. Restart replication on the subscriptions.
For instructions, see section *Starting and ending replication on subscriptions* in the *InfoSphere Change Data Capture Management Console, Administration Guide*.

Results

When restarted, replication on the subscriptions resumes and the databases are re-synchronized. Replication continues normally.

InfoSphere CDC instance fails

If the InfoSphere CDC instance fails, replication on the subscriptions ends also. To recover, proceed as follows:

1. Restart the InfoSphere CDC instance.
For instructions, see the section “Starting and stopping InfoSphere CDC” on page 40.
2. Restart replication on the subscriptions.
For instructions, see section *Starting and ending replication on subscriptions* in the *InfoSphere Change Data Capture Management Console, Administration Guide*.

Results

When restarted, replication on the subscriptions resumes and the databases are re-synchronized. Replication continues normally.

If this failure happens, the solidDB server continues to process transactions until it reaches the limit specified by the **LogReader.MaxLogSize** parameter.

solidDB server in HA mode (HotStandby) fails

The following sections describe failure scenarios in a solidDB HotStandby configuration.

Primary solidDB server fails

If the primary solidDB server fails, a high availability manager, like the High-Availability Controller (HAC), performs a failover to the secondary solidDB server as a standard procedure. If the 2-Safe protocol is used, the database and log states are fully preserved. The applications perceive a failover time of less than one second.

- If the solidDB Universal Cache is configured as a write-only cache (the data is replicated only from the front-end to the back-end), the InfoSphere CDC instance reconnects automatically to the new primary, and replication continues.
- If the solidDB Universal Cache is configured as read-only or read-write cache, replication on the subscriptions ends. The subscription needs to be restarted with the Management Console or with the InfoSphere CDC command `dmstartmirror`.

For instructions, see section *Starting and ending replication on subscriptions* in the *InfoSphere Change Data Capture Management Console, Administration Guide*.

During the scenario above, the InfoSphere CDC instance is in operation all the time.

Tip: For more information on the HA (HotStandby) functionality and the High-Availability Controller (HAC), see the *IBM solidDB High Availability User Guide*.

Secondary solidDB front-end fails

In the case of secondary front-end failure, no manual intervention is needed.

If the secondary front-end fails, the secondary front-end node is recovered in a normal way that is specific to the installation (for example, automatically rebooted). HAC automatically performs the rest of the recovery. The failure is not visible to applications or to InfoSphere CDC instance.

Communication link between primary solidDB server and InfoSphere CDC for solidDB instance fails

If the communication link between the primary solidDB server and the InfoSphere CDC for solidDB instance fails, replication on the subscriptions ends also. However, the failure of the link alone is considered unlikely.

To recover, proceed as follows:

1. Restart the InfoSphere CDC instance.

For instructions, see the section “Starting and stopping InfoSphere CDC” on page 40.

2. Restart replication on the subscriptions.

For instructions, see section *Starting and ending replication on subscriptions* in the *InfoSphere Change Data Capture Management Console, Administration Guide*.

Results

When restarted, replication on the subscriptions resume and the databases are re-synchronized. Replication continues normally.

If this failure happens, the solidDB server continues to process transactions until it reaches the limit specified by the **LogReader.MaxLogSize** parameter.

Back-end server or back-end node fails

If the back-end server or the back-end node fails, replication on the subscriptions ends also. To recover, proceed as follows:

Procedure

1. Restart the back-end server and recover the database.
2. Restart the InfoSphere CDC instance.
3. Restart mirroring (replication) on the subscriptions.

Note: It may be possible to automate the above steps using the back-end database-product-specific tools and procedures.

Results

When restarted, the replication resumes and the databases are re-synchronized. Replication continues normally.

If this failure happens, the solidDB front-end continues to process transactions until it reaches the limit specified by the **LogReader.MaxLogSize** parameter.

Back-end primary server fails

In the case of a failure in the back-end primary server, or if the whole back-end node fails, the recovery must be handled according to rules and tools of the back-end product in question. solidDB server does not offer any means to correct the situation.

Once the back-end server is running as a new primary, an exact copy of the InfoSphere CDC instance is restarted at the surviving node. The subscription has to be re-configured, with InfoSphere CDC tools, to reconnect the CDC instances in question. The new subscription will have to continue from a full refresh (in both directions) before the mirroring can start.

In some cases, the state of the subscription replication may be lost, and a full refresh is needed.

6 Troubleshooting

This section provides instructions and guidelines on how to prevent or troubleshoot common problems while configuring or using the solidDB Universal Cache.

Initial connections are not successful

The components for the solidDB Universal Cache must be installed and configured in the order described in 2, “Installing solidDB Universal Cache,” on page 13 and 3, “Configuring solidDB Universal Cache,” on page 15. Review the steps below and ensure that the installation and configuration steps were followed.

Installation order

- front-end (solidDB) and back-end data servers
- InfoSphere CDC for solidDB and InfoSphere CDC for the back-end data server
- Access Server
- Management Console

Important: At the end of the InfoSphere CDC for solidDB and InfoSphere CDC for the back-end data server installations, the installer prompts you to start the Configuration Tool for creating a new InfoSphere CDC instance. Do not select to start the configuration tool unless you have configured the corresponding data server according to the instructions in “Configuring solidDB” on page 15 and “Configuring back-end data server” on page 16.

Configuration order

- front-end and back-end data servers
- InfoSphere CDC instances
- Access Server and Management Console

Dependencies between components used in replication

To set up replication between databases, you need define and create various entities and components which are dependent on each other. These entities and components must be created in the following order and modified or deleted in the reverse order. For more details and instructions, see the *InfoSphere Change Data Capture Management Console, Administration Guide*.

1. Databases
2. InfoSphere CDC instances
3. Datastores
4. Subscriptions
5. Table mappings

Making changes to replication subscriptions

If you need to make changes to your replication subscriptions, you must first end replication on your subscriptions. For more details and instructions, see section *Ending replication on a subscription* in the *InfoSphere Change Data Capture Management Console, Administration Guide*.

Configuring InfoSphere CDC for solidDB fails - solid.jdbc.SolidDriver not found (Windows)

When creating or modifying a InfoSphere CDC for solidDB instance in Windows operating systems, the Configuration Tool displays an error message *Cannot save changes, solid.jdbc.SolidDriver or solid.jdbc.SolidDriver*.

Causes

- The CLASSPATH variable is set only in the "User variables" while InfoSphere CDC for solidDB is trying to read it from the "System variables". This happens if you choose "Local System account" as the service account.
- The CLASSPATH variable is not queried from the system at the time of the new instance creation, in the Configuration Tool.

If you set the CLASSPATH variable while the Configuration Tool > New Instance panel is open, InfoSphere CDC for solidDB may not be able to use the new setting.

- If the User CLASSPATH was updated, you have to restart the configuration tool for the change to have effect.
- If the System CLASSPATH was updated, you have to restart your computer.

Workaround

1. Exit the Configuration Tool:
 - a. Click **OK** to exit the error message.
 - b. Click **Cancel** to exit the New instance window.
 - c. Click **OK** to cancel without saving.
 - d. Close the Configuration Tool window.

This ensures that the cleanup of the failed instance is performed properly.

2. Continue with one of the two alternatives below:
 - Create the InfoSphere CDC for solidDB instance with the 'This account' option, using the same Windows user account that was used for installing solidDB.
 - Add the solidDB JDBC Driver to the System CLASSPATH variable manually, restart your computer, and create the InfoSphere CDC for solidDB instance.

InfoSphere CDC for solidDB hangs

InfoSphere CDC for solidDB instances may occasionally hang. This may cause the replication to stop without any visible indication.

To revert to normal operation:

1. Restart instances at both ends of the subscription.

In Windows operating systems, you can also restart the corresponding services using the **Control Panel > Administrative Tools > Services** dialog.
2. Restart replication (mirroring) on the subscription in the Management Console or with the dmstartmirror command in the command prompt.

7 InfoSphere CDC for solidDB (End-User Documentation)

About this section

This section contains detailed instructions for how to install and configure the IBM InfoSphere Change Data Capture for IBM solidDB component used in solidDB Universal Cache and InfoSphere CDC Replication. This section also contains the InfoSphere CDC specific commands as well as other reference information.

When setting up solidDB Universal Cache or InfoSphere CDC Replication, follow the system level installation and configuration instructions, referring to this section as necessary.

In this section, the term InfoSphere CDC is used for referring to InfoSphere CDC for solidDB.

About InfoSphere CDC for IBM solidDB

IBM InfoSphere Change Data Capture (InfoSphere CDC or CDC) is a replication solution that allows you to replicate data to or from supported databases. It can also receive replicated data from supported databases based on table mapping details configured in Management Console.

InfoSphere CDC allows you to maintain a replicated database that can be used to reduce processing overheads and network traffic. Replication can be carried out continuously or periodically on a net change basis. When data is transferred from a source server, it can be re-mapped or transformed in the target environment.

System requirements

Software requirements

See solidDB Web pages at <http://www-01.ibm.com/software/data/soliddb/> for supported platforms and operating systems.

Hardware requirements

Table 4. Hardware requirements

RAM	Disk space
500 MB	220 MB
	For installation files, data queues, and log files.

Port requirements

InfoSphere CDC requires that you allocate a set of ports for communications with client workstations running Management Console and other servers. The ports must be accessible through firewalls, although you do not require access to the internet.

Table 5. Port requirements

Protocol	Default port	Purpose
TCP	11101	<p>Accepts connections from:</p> <ul style="list-style-type: none"> • Management Console • Other installations of InfoSphere CDC as a source of replication • Command line utilities <p>For more information on how to install Management Console, see the <i>Management Console Installation Guide</i>.</p>

Required database, user accounts, and schemas

Creating a solidDB database

When you configure InfoSphere CDC, you are prompted for the host name and port number of the solidDB server you want InfoSphere CDC to connect to and replicate data. Before installing InfoSphere CDC, ensure that this solidDB database exists and that you have created and set up a database user that has access to it.

Setting up a solidDB account with SYS_ADMIN_ROLE privileges

Create and set up a Sybase user and assign DBA privileges to this user. For InfoSphere CDC to connect to your solidDB database, you need to create a solidDB user account and assign SYS_ADMIN_ROLE privileges to this user. When you configure InfoSphere CDC, you are prompted for the host name and port number of the solidDB server you want InfoSphere CDC to connect to and the user name and password of the solidDB user that has access to this database.

Creating a solidDB schema

Create a schema or choose an existing schema for your InfoSphere CDC database metadata tables. You will have to specify this schema when you configure InfoSphere CDC.

Setting up a Windows user account for InfoSphere CDC

If you are installing InfoSphere CDC on a Windows system, you must set up a new, or decide on an existing Windows account that you will use to install, configure, or upgrade InfoSphere CDC.

Setting up a Linux® or UNIX user account for InfoSphere CDC

If you are installing InfoSphere CDC on a Linux or UNIX system, you must set up a new, or decide on an existing Linux or UNIX account that you will use to install, configure, or upgrade InfoSphere CDC. You can install InfoSphere CDC in the directory of your choice, however, it must be owned by the Linux or UNIX account.

Single byte and multibyte character support

InfoSphere CDC supports replication of both single byte and multibyte character sets.

Considerations for replicating multibyte character sets

InfoSphere CDC supports the replication of multibyte character sets (MBCS) such as Japanese or Chinese, which cannot be represented in a single-byte. The most common MBCS implementation is double-byte character sets (DBCS).

Converting multibyte character sets: When you set an encoding conversion in Management Console, InfoSphere CDC applies the data to the target database in the exact form it was received.

User exits and multibyte character sets: Java™ class user exits in InfoSphere CDC support multibyte character sets (MBCS). Multibyte character sets are converted to Java strings (UTF-16).

Implications for multibyte character support: The specification for MBCS dictates that data will be applied as is to the mapped column on the target system when you have configured a specific translation. This is possible when the database has a single-byte character set configured (regardless of the actual character set of the data) but this cannot be assured when the character set is multibyte.

InfoSphere CDC will respect the mappings and apply the data according to the configuration set. There will be no validation that the character set can be inserted correctly into the column. You must be aware of the character sets on the database and select the appropriate values when selecting character set translations for their data.

For solidDB wide character data type columns (WCHAR, WVARCHAR, and so on) the character set translation must be set to UTF-16BE. For details on how to set the character set translation, see section *Setting data translations on column mappings* in the *InfoSphere Change Data Capture Management Console, Administration Guide*.

Single byte character support

InfoSphere CDC performs single byte character support (SBCS) code page conversions transparently. This means that you do not have to be aware of the code pages that are being used on each system. InfoSphere CDC is able to perform the conversions automatically by examining user configuration parameters.

Installing InfoSphere CDC

This section provides step-by-step instructions on how to install InfoSphere CDC.

If you are installing a Fix Pack, you must already have an installation of InfoSphere CDC in order to successfully complete the installation.

Installing InfoSphere CDC using an interactive installation

You can install InfoSphere CDC on a Windows server or an UNIX or Linux server.

To install InfoSphere CDC (Windows) Procedure

1. Double-click on the installation file. The InfoSphere CDC installation wizard opens.
2. Click **Next**.
3. If you agree to the license terms, select **I accept the terms in the license agreement** and then click **Next**.
4. Select the folder where you want to install InfoSphere CDC and click **Next**.

5. If you have a previous installation of InfoSphere CDC, the installation will prompt you to upgrade the installation. Click **OK** to upgrade the installation.
6. Select the location for the product icons and click **Next**.
7. Review the installation summary and click **Install**.
8. Optionally, select **Launch Configuration Tool** to launch the configuration tool after the installation. The configuration tool allows you to add an instance of InfoSphere CDC.
9. Click **Done** to exit the installation.

To install InfoSphere CDC (Unix and Linux)

About this task

Note: If you have X-Windows installed, the installation program will launch the configuration tool in a graphical environment. The configuration process is similar to Windows except you do not have to start and stop instances.

Procedure

1. Log on to the account you set up for InfoSphere CDC.
2. Copy the InfoSphere CDC installation file for your Linux platform.
3. Make the installation program executable.
4. Run the installation program by typing the name of the installation file.
5. Press Enter on the **Introduction** screen to display the license agreement. Follow the instructions on the screen to navigate through the license agreement.
6. To accept the license agreement, type 1.
7. Type the absolute path to your installation directory or press Enter to accept the default.

Note: The directory that you specify must be owned by the account you are using for the installation. If the installation program cannot create the directory, you are prompted to specify a different directory.

8. Review the installation summary. Press Enter to start the installation.
9. After completing the installation, InfoSphere CDC gives you the option of launching the configuration tool for InfoSphere CDC.
10. Type 1 to launch the configuration tool.

Installing InfoSphere CDC using a silent installation

A silent installation allows you to automatically install InfoSphere CDC by specifying a command with various parameters. You can use this type of installation method for large-scale deployments of InfoSphere CDC by embedding the silent installation command in a script.

To perform a silent installation of InfoSphere CDC (UNIX and Linux)

Procedure

1. Log on to the account you set up for InfoSphere CDC.
2. Copy the InfoSphere CDC installation file.
3. Make the installation program executable.
4. Install InfoSphere CDC and generate a response file with the following command:

```
<setup.bin> -r <response-file>
```

- On another system, perform the silent installation by running the following command:

```
<setup.bin> -i silent -f <response-file>
```

 where:
 - <response-file> is the full path to the installation file.

Configuring InfoSphere CDC (Windows)

After installing InfoSphere CDC, the installation program launches a configuration tool. The configuration tool allows you to configure InfoSphere CDC for your environment. You must configure InfoSphere CDC before you can start replication.

Configuring InfoSphere CDC instances (Windows)

You can add, edit, or delete an instance of InfoSphere CDC. Use the InfoSphere CDC configuration tool to work with instances.

To add a new instance of InfoSphere CDC (Windows) Before you begin

If you are configuring the first instance of InfoSphere CDC after installation, you can proceed to Step 3 of this procedure.

Procedure

- At the command prompt, launch the configuration tool by issuing the following command in the specified directory:

```
\<InfoSphere CDC Installation Directory>\bin\dmconfigurets
```
- At the welcome message, click **OK** to continue.
- On the **IBM InfoSphere CDC New Instance** dialog box, you can configure the following options in the **Instance** area:

Option	Description
Name	Type a name for your InfoSphere CDC instance. This name must be unique.
Server Port	Type the port number which InfoSphere CDC uses for communication with client workstations running Management Console and other servers. Note: This port number cannot be used by other applications installed on the same server. You will use this port number when specifying access parameters for your datastore in the Access Manager perspective in Management Console. InfoSphere CDC displays a default TCP/IP port of 11101. For more information, see your Management Console documentation. Note: If you install several instances on the same node, the port number for each instance must be unique.
Auto-Discovery Port	Select the box and type the UDP port number that is used for auto-discovery broadcasts sent from Access Server. For more information about auto-discovery, see your Management Console documentation.

Option	Description
Maximum Memory Allowed	Type the maximum amount of RAM you want to allocate for InfoSphere CDC. You must allocate a minimum of 64 MB for each instance you configure. By default, there is 512 MB of RAM allocated for a 32 bit instance and 1024 MB of RAM allocated for a 64 bit instance.
Bit-Version	Select the bit-version of your database by selecting one of the following options: <ul style="list-style-type: none"> • 32 bit • 64 bit <p>These options are not enabled if you are installing InfoSphere CDC on a 32-bit server.</p>

4. In the **Windows Service** area, you can specify the account that will be used to start InfoSphere CDC services. Select one of the following options:

Option	Description
Local System account	Start InfoSphere CDC services through the local system administrator account.
This account	Start InfoSphere CDC services through the specified user account. <p>The account must be specified in the format <i><domain>\<user name></i>, where <i><domain></i> is the name of a domain in your environment, and <i><user name></i> is a valid login user name in the specified domain. If your computer is not part of a domain, you can specify <i><computer name>\<user name></i>.</p> <p>In the Password and Confirm Password boxes, type the password currently associated with the selected Windows user account. If you change the password for the Windows user account after installing InfoSphere CDC, you will have to use the Windows Services dialog to change the password currently set for each InfoSphere CDC service.</p>

5. In the **Database** area, you can configure access to the database that contains the tables for replication. To complete this step, you will require system administrator privileges. You can then add a datastore in the Access Manager perspective in Management Console and provide users access to this database. For more information, see your Management Console documentation.

Option	Description
User name	Type the user name for the specified database.
Password	Type the password for the specified database.

Option	Description
Metadata Schema	Select the schema for the database that will be used for the InfoSphere CDC metadata tables. As a default, the user name entered above is used. You can specify any schema except those in use by other installed instances of InfoSphere CDC for the given database. You must set up or decide on this schema as part of the installation prerequisites. Note: Make sure to use UPPERCASE letters for the metadata schema. By default, all the schema names (catalog names) in solidDB are in uppercase.
Advanced	The Advanced button enables you to modify configuration parameters for the solidDB JDBC driver. For more information on the JDBC driver parameters, see the <i>IBM solidDB Programmer Guide</i> .

- In the **Server** area, you can configure the solidDB server that you want to replicate data to or from and which contains all of the tables for replication. You can configure either single server or HA configuration (HotStandby).

Option	Description
Single server	Type the host name and port number for the specified solidDB server.
HA Configuration (HotStandby)	Type the host names and port numbers for the specified Primary and Secondary solidDB servers.

- Click **OK** to save your configuration settings for the InfoSphere CDC instance.
- Click **Apply** to save your changes in the configuration tool.

What to do next

After you complete the configuration, you can start InfoSphere CDC.

To edit an instance of InfoSphere CDC (Windows) Procedure

- Stop InfoSphere CDC if it is started by using the `dmshutdown` command.
- Launch the configuration tool at the command prompt by issuing the following command in the specified directory:
`\<InfoSphere CDC Installation Directory>\bin\dmconfigurets`
- In the **Instances** area, select the instance that you want to modify and click **Stop** if the instance is started.
- In the **Instances** area, select an instance and click **Edit**.
The **InfoSphere CDC Edit Instance** dialog opens.
- You can modify any of the values on this dialog box that you specified when adding an instance.
- Click **Apply** to save your changes and then click **Close**.
The configuration tool will modify the instance.

7. In the **Instances** area, select the instance that you modified and click **Start** to start the instance.

To delete an instance of InfoSphere CDC (Windows) Procedure

1. Stop InfoSphere CDC if it is started by using the `dmshutdown` command.
2. At the command prompt, launch the configuration tool by issuing the following command in the specified directory:
`\<InfoSphere CDC Installation Directory>\bin\dmconfigurets`
3. In the **Instances** area, select the instance that you want to delete and click **Stop** if the instance is started.
4. In the **Instances** area, select an instance and click **Delete**.
5. Click **Yes** to permanently delete the instance.

Configuring InfoSphere CDC (UNIX and Linux)

After installing InfoSphere CDC, the installation program launches a configuration tool. The configuration tool allows you to configure InfoSphere CDC for your environment. You must configure InfoSphere CDC before you can start replication.

Configuring InfoSphere CDC instances (UNIX and Linux)

You can add, edit, or delete an instance of InfoSphere CDC. Use the InfoSphere CDC configuration tool to work with instances.

To add a new instance of InfoSphere CDC (UNIX and Linux) Before you begin

If you are configuring the first instance of InfoSphere CDC after installation, you can proceed to Step 4 of this procedure.

Procedure

1. At the command prompt, launch the configuration tool by issuing the following command in the specified directory:
`/<InfoSphere CDC Installation Directory>/bin/dmconfigurets`
2. At the welcome message, press **Enter** to continue.
3. Type `2` and press **Enter** to add a new instance of InfoSphere CDC.
4. Type a name for your InfoSphere CDC instance and press **Enter**. The instance name must be unique.
5. Type the port number which InfoSphere CDC uses for communication with client workstations running Management Console and other servers. InfoSphere CDC displays a default port of 11101. Press **Enter**.

Note: This port number cannot be used by other applications installed on the same server. You will use this port number when specifying access parameters for your datastore in the Access Manager perspective in Management Console. For more information, see your Management Console documentation.

Note: If you install several instances on the same node, the port number for each instance must be unique.

6. Type your auto-discovery UDP port number or press **Enter** to use the default value of DISABLE. This port number is used for auto-discovery broadcasts sent from Access Server. For more information about auto-discovery, see your Management Console documentation.
7. Type the maximum amount of RAM you want to allocate for InfoSphere CDC. You must allocate a minimum of 64 MB for each instance you configure. By default, there is 512 MB of RAM allocated for a 32 bit instance and 1024 MB of RAM allocated for a 64 bit instance.
8. Select the solidDB server configuration type you want to configure.

Option	Description
Single server	Type 1 and press Enter .
HA Configuration (HotStandby)	Type 2 and press Enter .

9. Type the host name and port number according to your configuration type.

Option	Description
Single server	<ol style="list-style-type: none"> 1. Type the host name for the specified server and press Enter. 2. Type the port number for the specified server and press Enter. Default is 1964.
HA Configuration (HotStandby)	<ol style="list-style-type: none"> 1. Type the host name for the specified Primary server and press Enter. 2. Type the port number for the specified Primary server and press Enter. Default is 1964. 3. Type the host name for the specified Secondary server and press Enter. 4. Type the port number for the specified Secondary server and press Enter. Default is 1964. Note: The default port number for Primary and Secondary is the same as it is assumed that the Primary and Secondary are located on different nodes. If, for example, for evaluation purposes, your Primary and Secondary servers are located on the same node, the default port number for both cannot be the same.

10. Configure advanced parameters (JDBC parameters) as necessary.

Option	Description
Use default settings	Type n and press Enter .
Modify settings	<ol style="list-style-type: none"> 1. Type y and press Enter 2. Enter the parameter settings using the syntax <parameter>=<value>;<parameter>=<value>;... <p>Note: In HA setup, the parameter solid_tf_level is by default set to 'CONNECTION'.</p>

11. Type the user name for the specified database and press **Enter**.
12. Type the password for the specified database and press **Enter**. The configuration tool will now search the database for schemas.

13. Type the number that corresponds to the metadata schema that you would like to use and press **Enter**.
14. Type the path to the directory that will be used for bulk inserts into the database. Press **Enter**. Both your solidDB database and InfoSphere CDC must have read and write permissions for this directory.

Notes:

- You should use a different directory for each instance of InfoSphere CDC.
 - This directory may contain database tables for replication. You should take this into consideration when determining user access to this directory.
15. The configuration tool creates the InfoSphere CDC instance and prompts you to start the instance. Type **y** to start the instance.

Note: The configuration tool will prompt you if your configuration is about to overwrite the metadata for an existing instance.

To edit an instance of InfoSphere CDC (UNIX and Linux) Procedure

1. Stop InfoSphere CDC if it is started by using the `dmshutdown` command.
2. Launch the configuration tool by issuing the following command in the specified directory:
`/<InfoSphere CDC Installation Directory>/bin/dmconfigurets`
3. Type **1** and press **Enter** to list the installed instances of InfoSphere CDC. Record the name of the instance you want to modify.
4. Type **3** and press **Enter** to modify an instance of InfoSphere CDC.
5. Type the instance name that you want to modify and press **Enter**.
The configuration tool allows you to edit a number of values that you specified when adding an instance.
6. After making your changes, type **5** and press **Enter** to apply your changes and return to the main menu. Type **6** and press **Enter** to discard your changes.

To delete an instance of InfoSphere CDC (UNIX and Linux) Procedure

1. Stop InfoSphere CDC if it is started by using the `dmshutdown` command.
2. Launch the configuration tool by issuing the following command in the specified directory:
`/<InfoSphere CDC Installation Directory>/bin/dmconfigurets`
3. Type **1** and press **Enter** to list the installed instances of InfoSphere CDC. Record the name of the instance you want to delete.
4. Type **4** and press **Enter** to delete an instance of InfoSphere CDC.
5. Type the instance name that you want to delete and press **Enter**.

Starting and stopping InfoSphere CDC

This section provides step-by-step instruction on how to start and stop InfoSphere CDC instances.

Starting InfoSphere CDC

When you install InfoSphere CDC on a supported Windows server, you can start it manually after the initial configuration. Starting InfoSphere CDC starts the services in Windows. The services will automatically start after a reboot.

When you install InfoSphere CDC on a supported Linux server, you can issue a command to start it. After installing InfoSphere CDC, start it so that you can create a datastore for this instance in Management Console.

To start InfoSphere CDC (Windows)

Procedure

1. At the command prompt, launch the configuration tool by issuing the following command in the specified directory:

```
\<InfoSphere CDC Installation Directory>\bin\dmconfigurets
```

2. In the **Instances** area, select the instance that you want to start and click **Start**.
The configuration tool starts the instance of InfoSphere CDC.

What to do next

You can also use the Windows Services dialog to start and stop InfoSphere CDC services.

To start InfoSphere CDC (UNIX and Linux)

Procedure

Depending on the operating system you are running InfoSphere CDC, issue one of the following start commands:

- `dmts32 - I <instance_name>`
- `dmts64 - I <instance_name>`

Stopping InfoSphere CDC

It may be necessary to stop InfoSphere CDC when you want to change the configuration settings using the InfoSphere CDC configuration tool.

On Windows, stopping InfoSphere CDC stops the services in Windows. The services will automatically start again after a reboot.

On UNIX and Linux, you can issue a command to stop InfoSphere CDC. Use the command prior to taking a server or database offline for maintenance purposes or for upgrading InfoSphere CDC.

To stop InfoSphere CDC (Windows)

Procedure

1. Launch the configuration tool by issuing the following command in the specified directory:

```
/<InfoSphere CDC Installation Directory>/bin/dmconfigurets
```

2. In the **Instances** area, select the instance that you want to start and click **Stop**.
The configuration tool starts the instance of InfoSphere CDC.

What to do next

You can also use the Windows Services dialog to start and stop InfoSphere CDC services.

To stop InfoSphere CDC (UNIX and Linux) Procedure

1. End replication on all subscriptions in Management Console. For more information on how to end replication on subscriptions, see your Management Console documentation.
2. Depending on how you want to stop InfoSphere CDC, issue one of the following stop commands:

Option	Description
<code>dmshutdown -I <instance_name></code>	Use this command to gracefully shut down InfoSphere CDC. If you have multiple active InfoSphere CDC installations on the same Linux server, and you want to shut them all down, run this command from the installation directory for each InfoSphere CDC instance.
<code>dmterminate -I <instance_name></code>	Use this command to terminate all InfoSphere CDC processes for all instances running on a Linux server. Use this command when you cannot completely shut down InfoSphere CDC using the <code>dmshutdown</code> command.

Enabling SQL statements in Management Console

InfoSphere CDC lets you execute SQL statements after it applies a table-level clear or refresh operation to a target table. You can specify SQL statements in the **Additional SQL** dialog box in Management Console. By default, this feature is disabled in InfoSphere CDC for security reasons. You can enable this feature by creating a table called `TS_SQL_EXECAUTH` in the database where you installed InfoSphere CDC. The structure of the table is unimportant, and you must create the table using the same schema as the metadata tables during the configuration of InfoSphere CDC. For more information about specifying SQL statements in Management Console, see "Specifying SQL to control refresh operations" in your Management Console documentation.

To enable SQL statements in Management Console Procedure

1. Locate the database on the target server that you created for InfoSphere CDC. Depending on how you are using InfoSphere CDC, this is the database you want InfoSphere CDC to replicate to or from.

Note: During installation, InfoSphere CDC places metadata tables in the database necessary for InfoSphere CDC processes.

2. If you want to enable the specification of SQL statements, create a table named `TS_SQL_EXECAUTH` in the database.

Note: The table can have any structure and must be created in the schema you specified when you configured InfoSphere CDC.

Datatypes supported by InfoSphere CDC

When you map source and target columns for replication in Management Console, you should know which data types are compatible. For more information on how to map tables for replication, see the "Mapping tables" section in your Management Console documentation.

Supported data types

This section identifies the data types that InfoSphere CDC can replicate.

- bigint
- binary
- blob
- char
- clob
- date
- decimal
- double precision
- float
- integer
- long varbinary
- long varchar
- nchar
- nclob
- numeric
- nvarchar
- real
- smallint
- time
- timestamp
- tinyint
- varbinary
- varchar

For more details on the supported data types, see the *IBM solidDB SQL Guide*.

Supported mappings

This section indicates the supported Management Console mappings for supported data types.

Published data types	Supported mapping
bigint	Any numeric, binary or LOB data type
binary	Any binary or LOB data type
blob	Any binary or LOB data type
char	Any character, variable character, CLOB, binary, or other LOB data type
clob	Any character, variable character, CLOB, binary, or other LOB data type

Published data types	Supported mapping
date	Any data type
decimal	Any numeric, binary or LOB data type
double precision	Any numeric, binary or LOB data type
float	Any numeric, binary or LOB data type
integer	Any numeric, binary or LOB data type
long varbinary	Any binary or LOB data type
long varchar	Any character, variable character, CLOB, binary, or other LOB data type
nchar	Any character, variable character, CLOB, binary, or other LOB data type
nclob	Any character, variable character, CLOB, binary, or other LOB data type
nvarchar	Any character, variable character, CLOB, binary, or other LOB data type
numeric	Any numeric, binary or LOB data type
real	Any numeric, binary or LOB data type
smallint	Any numeric, binary or LOB data type
tim	Any time data type
timestamp	Any date, time, or timestamp data types
tinyint	Any numeric, binary or LOB data type
varbinary	Any binary or LOB data type
varchar	Any character, variable character, CLOB, binary, or other LOB data type

InfoSphere CDC metadata tables

InfoSphere CDC maintains a set of tables that represent data about your current replication configuration. For various reasons, you may want to back up these tables, or perform other similar tasks. However, do not modify the contents of these tables unless requested to do so by your IBM representative.

The names of the metadata tables created by InfoSphere CDC are as follows:

- TS_AUTH

Note: For all users you added in the Access Manager perspective in Management Console, make sure you give GRANT SELECT privileges to the TS_AUTH metadata table. For more information on how to add users in the Access Manager perspective in Management Console. For more information Access Manager perspective, see your Management Console documentation.

- TS_BOOKMARK
- TS_CONFAUD

InfoSphere CDC creates and installs these tables in the database and schema that you specify during the configuration process.

If you want to specify SQL statements after InfoSphere CDC applies a table-level refresh or clear operation to the target table, then you need to create and maintain the TS_SQL_EXECAUTH table.

Commands for InfoSphere CDC

This section discusses the commands available with InfoSphere CDC. Using these commands you can control replication, manage your tables for replication, monitor replication, and perform various other tasks.

Using the InfoSphere CDC commands

You can issue InfoSphere CDC commands at a command line prompt or as part of a batch file or shell script. Commands are located in the `bin` directory of your InfoSphere CDC installation directory. Navigate to this directory to run the commands.

Note: To list the available flags for a command and a short description of each flag, type the name of the command at a command prompt with the `-?` flag and press **Enter**. For example, `dmterminate -?`.

Command formats

For each command, the following items of information are provided:

- **Syntax**—Identifies the name of the command and lists the command parameters.
- **Parameters**—Describes each parameter in the command and identifies the values that can be specified.
- **Result**—Indicates the values that are returned by the command if it is successful. These values can be useful for scripting. This section also specifies the information that is displayed on the screen, if any, as a result of executing the command.
- **Examples**—Provides one or more examples of invoking the command.

Parameter formats

Note the following conventions in the definition of the command parameters:

- Angle brackets (`< >`) indicate a **mandatory** parameter.
- Square brackets (`[]`) indicate an **optional** parameter. If you omit the parameter, InfoSphere CDC uses a default value.
- A vertical bar (`|`) separating one or more parameters indicate that only one of the parameters in the list can be used. When one or more vertical bars appear in a list of parameters that is enclosed by square brackets `[]`, the choices are limited to the parameters in the list, but you have the option to not specify any of the parameters.
- Ellipsis (`...`) means that a parameter or option can be repeated more than once.
- Unless otherwise noted, the commands apply to Linux and Windows operating systems.

Setting the TSINSTANCE environment variable

Before using the commands, you can set the `TSINSTANCE` environment variable to the name of your InfoSphere CDC instance.

After you set the `TSINSTANCE` environment variable, you no longer have to specify the instance name when issuing commands.

Windows platform

Issue the following command at the command prompt:

```
SET TSINSTANCE=<instance_name>
```

where:

- <instance_name> is the name of your InfoSphere CDC instance.

Linux platform

Issue the following command:

```
EXPORT TSINSTANCE=<instance_name>
```

where:

- <instance_name> is the name of your InfoSphere CDC instance.

Controlling replication commands

This section contains commands that control replication in InfoSphere CDC.

dmendreplication - End Replication

Use this command to end refresh or mirroring on the specified subscriptions. This command exits after it has successfully ended replication on the specified subscriptions.

Syntax

```
dmendreplication -I <instance_name> [-c|-i] [-A|-s <subscription_names> ...> [-L <locale>]
```

Parameters

-I <instance_name>

Specifies the InfoSphere CDC instance for which you want to end replication. Alternatively, you can specify the TSINSTANCE environment variable in place of this value.

-c Specifies that InfoSphere CDC ends replication in a controlled manner. This option lets InfoSphere CDC complete all in-progress operations and apply pending changes to the target table. If you omit both the **c** and **i** options, InfoSphere CDC assumes **c** by default.

-i Specifies that InfoSphere CDC ends replication immediately. This option interrupts any current operations. InfoSphere CDC does not apply pending changes to the target table.

-A Specifies that InfoSphere CDC ends replication on all subscriptions.

-s <subscription_names>

Specifies that InfoSphere CDC ends replication on the indicated subscriptions. List the subscriptions if you specify more than one.

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

Examples

```
dmendreplication -I myinstance -c -s Finance
```

InfoSphere CDC ends replication in a controlled manner for the **Finance** subscription for the specified instance.

dmrefresh - Refresh Subscription

Use this command to refresh the specified subscriptions. When you refresh a subscription, InfoSphere CDC ensures that the target tables are synchronized with the source tables. Typically, you would refresh target tables when you have set the replication method to **Refresh** on your tables.

However, you can also refresh target tables that have a replication method set to Mirror. When you refresh a table configured for mirroring, InfoSphere CDC refreshes the target table so that it is synchronized with the source table and then sets a log position in the journal as the starting point for mirroring.

This command exits after it has successfully refreshed the specified subscriptions. If you terminate this program while it is still running, InfoSphere CDC ends replication immediately for the specified subscriptions.

Syntax

```
dmrefresh -I <instance_name> [-a|-f] [-A|-s <subscription_names> ...] [-L <locale>]
```

Parameters

-I <instance_name>

Specifies the InfoSphere CDC instance for which you want to refresh one or more subscriptions. Alternatively, you can specify the TSINSTANCE environment variable in place of this value.

-a Specifies that InfoSphere CDC refreshes all target tables in the subscription.

-f Specifies that InfoSphere CDC refreshes only target tables that are flagged for refresh. If you omit both the **-a** and **-f** options, InfoSphere CDC assumes **-f** by default.

-A Specifies that InfoSphere CDC refreshes all subscriptions.

-s <subscription_names>

Specifies that InfoSphere CDC refreshes the indicated subscription. List the subscriptions if you specify more than one.

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

Examples

```
dmrefresh -I new_instance -a -s Finance
```

InfoSphere CDC refreshes all target tables in the **Finance** subscription.

dmstartmirror - Start Mirroring

Use this command to start mirroring on the specified subscriptions. This command starts mirroring for any table with a replication method of **Mirror** and a status of **Refresh** or **Active**. Before starting mirroring, all tables with a replication method of **Mirror** and a status of **Refresh** are initially refreshed on the subscription. If you start continuous mirroring, this command exits after it has successfully started mirroring on the specified subscriptions. If you start net-change mirroring, this command exits after net change mirroring has completed on the specified subscriptions.

Syntax

```
dmstartmirror -I <instance_name> [-c|-n] <-A|-s <subscription_names> ...>
[-L <locale>]
```

Parameters

-I <instance_name>

Specifies the InfoSphere CDC instance for which you want to start mirroring. Alternatively, you can specify the TSINSTANCE environment variable in place of this value.

-c Specifies that InfoSphere CDC starts continuous mirroring. If you omit both the c and n options, InfoSphere CDC assumes c by default.

-n Specifies that InfoSphere CDC starts net change mirroring.

-A Specifies that InfoSphere CDC starts mirroring all subscriptions.

-s <subscription_names>

Specifies that InfoSphere CDC starts mirroring the indicated subscription. List the subscriptions if you specify more than one.

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

Examples

```
dmstartmirror -I myinstance c -s Finance
```

InfoSphere CDC starts continuous mirroring for the **Finance** subscription.

Database transaction log commands

This section contains commands that help you manage your database transaction log or bookmarks.

dmdecodebookmark - Display Verbose Bookmark Information

Use this command to display verbose information about a bookmark.

Syntax

```
dmdecodebookmark -I <instance_name> (-b | -f) [-d] [-L <locale>]
```

Parameters

-I <instance_name>

The name of the InfoSphere CDC instance. You can set the TSINSTANCE environment variable to the name of your InfoSphere CDC instance. After this is complete, you no longer have to specify the instance when issuing commands.

-b <bookmark>

The bookmark as a hexadecimal-encoded string.

-f <bookmark_file>

The bookmark file as a binary file.

[-d] <database_version>

The database and version that generated the bookmark specified, if the bookmark was generated by an earlier version of InfoSphere CDC.

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

Examples

```
dmdecodebookmark -f bookmark.txt
```

InfoSphere CDC displays information about the bookmark.txt file.

dmsetbookmark - Set Bookmark

Use this command to change the scraping point for a subscription.

Syntax

```
dmsetbookmark -I <instance_name> -s <subscription_name> (-b | -l | -f | -t) [-a] [-L <locale>]
```

Parameters

-I <instance_name>

The name of the InfoSphere CDC instance. You can set the TSINSTANCE environment variable to the name of your InfoSphere CDC instance. After this is complete, you no longer have to specify the instance when issuing commands.

-s <subscription_name>

The name of the subscription for which InfoSphere CDC sets a bookmark.

-b <bookmark>

Specifies the bookmark which determines the point in the database log where you want InfoSphere CDC to resume mirroring. The next time InfoSphere CDC mirrors, it will scrape at the given position. The bookmark is a hex encoded string that is obtained from the dmshowbookmark command.

-l <bookmark>

Bookmark indicating the new scraping point. The bookmark is a string obtained from the `dmdecodebookmark` command. For more information, see “`dmdecodebookmark - Display Verbose Bookmark Information`” on page 48.

-f <bookmark_file>

Specifies the binary file containing a bookmark which determines the point in the database log where you want InfoSphere CDC to resume mirroring. The next time InfoSphere CDC mirrors, it will scrape at the given position. The bookmark file is a binary file that stores the position.

-t <datetime>

Specifies the next time you want InfoSphere CDC to resume scraping at an entry logged around the given date and time.

-a Sets all tables in the subscriptions (except for parked tables) as active as of the new scraping point.

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails

Examples

```
dmsetbookmark -I MYINSTANCE -b 2FC5GJHKLKSLJKL458K9K809IK9  
-s FINANCE
```

InfoSphere CDC sets a bookmark position on the **Finance** subscription for the specified instance. This command specifies that mirroring will resume at the indicated position in the database log.

dmshowbookmark - Display Bookmark Information

Use this command to view the latest replication position for a subscription committed on the target.

Syntax

```
dmshowbookmark -I <instance_name> -s <subscription_name> [-f] [-v] [-L <locale>]
```

Parameters

-I <instance_name>

The name of the InfoSphere CDC instance. You can set the `TSINSTANCE` environment variable to the name of your InfoSphere CDC instance. After this is complete, you no longer have to specify the instance when issuing commands.

-s <subscription_name>

Specifies the name of the subscription for which the bookmark is displayed.

[-f] <file_name>

Specifies the name of the binary file for the bookmark output.

[-v]

Displays verbose information about a bookmark, including a hexadecimal-encoded string. The amount of information displayed depends on

the type and version of the source engine. The hexadecimal-encoded string is always displayed. It is a subset of what the `dmdecodebookmark` command displays. If not specified, only a hexadecimal-encoded string is displayed.

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

Examples

```
dmshowbookmark -I myinstance -s master
```

InfoSphere CDC shows bookmark information for the **master** subscription.

dmshowlogdependency - Show Log Dependency

Use this command to display information about the database logs that are used by InfoSphere CDC and are required for replication. Use this command to implement a log retention policy. With this command you can display the following information:

- A list of database logs that are required for the specified instance.
- A list of database logs that are currently missing for the specified instance.
- The earliest open transaction in the database log for the specified instance.
- The database log that the specified instance of InfoSphere CDC is currently reading on the source.
- The database log for the subscription that the specified instance of InfoSphere CDC is currently applying on the target.

Syntax

```
dmshowlogdependency -I <instance_name> (-c | -i | -t | -l)
(-s <subscription_name> | -A) [-v] [-L <locale>]
```

Parameters

-I <instance_name>

The name of the InfoSphere CDC instance. You can set the `TSINSTANCE` environment variable to the name of your InfoSphere CDC instance. After this is complete, you no longer have to specify the instance when issuing commands.

- c** Checks and displays a list of archived database logs that are currently missing for the specified instance and are required for replication. These logs must be restored before you start replication with InfoSphere CDC. If no logs are missing, the output of this command displays "OK". You can parse the output in a script or batch file.
- i** Displays the complete list of database logs that are required for the specified instance. These logs are required to start replication, and they contain data that has not yet been applied to the target.
- t** Displays the current target database log for a subscription that the specified InfoSphere CDC instance is currently reading. This is the log containing the current position confirmed by the target.

- l Displays the current source database log that the specified InfoSphere CDC instance is currently reading. This is the log containing the current scraping position.
- s <subscription_name>
Specifies the name of the subscription for which you want to display the target database log that InfoSphere CDC is currently reading. Use this parameter in conjunction with the -t parameter to display the target database log.
- A Specifies all subscriptions
- v Specifies verbose output (otherwise, the output is formatted for scripting).
- L <locale>
The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

Examples

```
dmshowlogdependency -I MyInstance
```

Managing tables for replication commands

This section contains commands that help you manage the tables that you want to replicate with InfoSphere CDC.

dmdescribe - Describe Source Tables

Use this command to send source table mapping changes over to the target.

This command exits after it has successfully described the specified subscriptions.

Syntax

```
dmdescribe -I <instance_name> [-A|-s <subscription_names> ...] [-L <locale>]
```

Parameters

- I <instance_name>
Specifies the InfoSphere CDC instance for which you want to send source table mapping changes over to the target. Alternatively, you can specify the TSINSTANCE environment variable in place of this value.
- A Specifies that InfoSphere CDC will send source table mapping changes made to all subscriptions over to the target.
- s <subscription_names>
Specifies that InfoSphere CDC sends source table mapping changes for the indicated subscriptions over to the target. List the subscriptions if you specify more than one.
- L <locale>
The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

Examples

```
dmdescribe -I new_instance -s Finance
```

InfoSphere CDC sends source table mapping changes in the **Finance** subscription over to the target for the specified instance.

dmflagforrefresh - Flag for Refresh

Use this command to flag a source table for refresh. When you flag a table for refresh, you are selecting the tables that you want to refresh at a future point in time. Use this procedure when you have selected **Refresh** as your replication method on a subscription.

Syntax

```
dmflagforrefresh -I <instance_name> -s <subscription_names>  
<-A|-t <schema>.<table> ...> [-L <locale>]
```

Parameters

-I <instance_name>

Specifies the name of the InfoSphere CDC instance. Alternatively, you can specify the TSINSTANCE environment variable in place of this value.

-s <subscription_names>

Specifies the name of the subscription. List the subscriptions if you specify more than one.

-A Specifies that InfoSphere CDC flags all source tables for refresh in the subscription.

-t <schema>.<table>

Specifies the name of a source table in the subscription that InfoSphere CDC flags for refresh. You must specify the table name in the format *schema.table*. List the tables if you specify more than one.

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

Examples

```
dmflagforrefresh -I myinstance -s Finance -A
```

InfoSphere CDC flags for refresh all source tables in the **Finance** subscription for the specified instance.

dmmarktablecapturepoint - Mark a table capture point on a source table

Use this command to mark a table capture point on a source table and move the table to active state. If you changed the table before executing this command, those changes will not be replicated.

Mark a table capture point on a source table when you want to override an existing position in the stream of changed data. This is possible when you have already synchronized (refreshed) your source and target tables using an application other than Management Console (for example, using the import or export capabilities of your database platform) and you know the point in time your source and target are synchronized with each other. InfoSphere CDC mirrors changes to the target table from the current position in the stream of changed data. This position is set by InfoSphere CDC when you select **Mirror (Change Data Capture)** after mapping your tables in the Map Tables wizard. If you want to override the position set by InfoSphere CDC, then you can manually mark a table capture point in Management Console. When you decide to start mirroring on the subscription, InfoSphere CDC identifies the position you have set as the point in time from which to capture and replicate database changes to the target.

Syntax

```
dmmarktablecapturepoint -I <instance_name> -s <subscription_names>  
<-A|-t <schema>.<table> ...> [-L <locale>]
```

Parameters

-I <instance_name>

Specifies the name of the InfoSphere CDC instance. Alternatively, you can specify the TSINSTANCE environment variable in place of this value.

-s <subscription_names>

Specifies the subscription name. List the subscriptions if you specify more than one.

-A Specifies that InfoSphere CDC overrides an existing position in the stream of changed data on all source tables in the subscription.

-t <schema>.<table>

Specifies the name of a source table in the subscription on which InfoSphere CDC marks a table capture point. You must specify the table name in the format *schema.table*. List the tables if you specify more than one.

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

Examples

```
dmmarktablecapturepoint -I myinstance -s Finance -A
```

InfoSphere CDC overrides an existing position in the stream of changed data on all source tables in the **Finance** subscription.

```
dmmarktablecapturepoint -I myinstance -s Finance -t myschema.mytable
```

InfoSphere CDC moves the specified table to active in the **Finance** subscription.

dmpark - Park Table

Use this command to park a source table. By parking a source table, you tell InfoSphere CDC that you do not want to capture changes for that particular table in a subscription. When you park a table, InfoSphere CDC does not replicate any subsequent changes you make on the source table, which may result in inconsistent source and target tables.

Note: Before you can park a source table, if you are mirroring the table to the target, then you need to end replication on the subscription. For more information, see “dmendreplication - End Replication” on page 46.

Syntax

```
dmpark -I <instance_name> -s <subscription_names> [-A] -t <schema>.<table> ...> [-L <locale>]
```

Parameters

-I <instance_name>

Specifies the name of the InfoSphere CDC instance. Alternatively, you can specify the TSINSTANCE environment variable in place of this value.

-s <subscription_names>

Specifies the subscription name. List the subscriptions if you specify more than one.

-A Specifies that InfoSphere CDC parks all source tables in the subscription.

-t <schema>.<table>

Specifies the name of a source table in the subscription that InfoSphere CDC parks. You must specify the table name in the format *schema.table*. List the tables if you specify more than one.

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

Examples

```
dmpark -I myinstance -s Finance -A
```

InfoSphere CDC parks all source tables in the **Finance** subscription.

dmreaddtable - Update Source Table Definition

Use this command to update the definition of a source table in InfoSphere CDC metadata. Run this command after you have changed the definition of a source table using your RDBMS.

Syntax

```
dmreaddtable -I <instance_name> [-A] -t <schema>.<table> ...> [-L <locale>]
```

Parameters

-I <instance_name>

Specifies the name of the InfoSphere CDC instance. Alternatively, you can specify the TSINSTANCE environment variable in place of this value.

-A Specifies that InfoSphere CDC updates definitions for all source tables that are available for replication.

-t <schema>.<table>

Specifies the name of a source table in the subscription for which InfoSphere CDC updates the definition. You must specify the table name in the format *schema.table*. List the tables if you want to specify more than one.

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

Examples

```
dmreadddtable -I new_instance -A
```

InfoSphere CDC updates definitions for all source tables that are available for replication.

dmreassigntable - Update Target Table Definition

Use this command to update the definition of a target table in InfoSphere CDC metadata. Run this command after you have changed the definition of a target table using your RDBMS.

Syntax

```
dmreassigntable -I <instance_name> -s <subscription_names>  
<-A|-t <schema>.<table> ...> [-L <locale>]
```

Parameters

-I <instance_name>

Specifies the name of the InfoSphere CDC instance. Alternatively, you can specify the TSINSTANCE environment variable in place of this value.

-s <subscription_names>

Specifies the InfoSphere CDC subscription that contains the table. List the subscriptions if you specify more than one.

-A Specifies that InfoSphere CDC updates definitions for all target tables in the subscription.

-t <schema>.<table>

Specifies the name of a target table in the subscription for which InfoSphere CDC updates the definition. You must specify the table name in the format *schema.table*. List the tables if you specify more than one.

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

Examples

```
dmreassigntable -I new_instance -s Finance -A
```

InfoSphere CDC updates definitions for all target tables in the **Finance** subscription.

dmsetreplicationmethod - Set Replication Method

Use this command to change the replication method for tables in a subscription. When running this command, InfoSphere CDC changes the status of any **Active** tables to **Refresh**.

Note: Before you run this command, you must end replication on the subscription.

Syntax

```
dmsetreplicationmethod -I <instance_name> <-r|-m> -s <subscription_names>  
<-A|-t <schema>.<table> ...> [-L <locale>]
```

Parameters

-I <instance_name>

Specifies the name of the InfoSphere CDC instance. Alternatively, you can specify the TSINSTANCE environment variable in place of this value.

-m Specifies that tables will use **Mirror (Change Data Capture)** as the replication method.

-r Specifies that tables will use **Refresh (Snapshot)** as the replication method.

-s <subscription_names>

Specifies the name of the subscriptions.

-A Specifies that all tables in the subscription will use the indicated replication method.

-t <schema>.<table>

Specifies the name of a source table in the subscription that will use the indicated replication method. You must specify the table name in the format *schema.table*. List the tables if you specify more than one.

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

Examples

```
dmsetreplicationmethod -I myinstance -r -s Finance -A
```

All tables in the **Finance** subscription will use **Refresh** as the replication method in the specified InfoSphere CDC instance.

```
dmsetreplicationmethod -I new_instance -m -s Finance -t acct.taxcodes
```

The source table *acct.taxcodes* in the **Finance** subscription will use **Mirror** as the replication method in the specified InfoSphere CDC instance.

Monitoring replication commands

This section contains commands that help you monitor replication in InfoSphere CDC:

dmclearevents - Clear Events

Use this command to delete events from the **Event Log** view in Management Console.

Syntax

```
dmclearevents -I <instance_name> [-S|-T|-B] <-A|-s <subscription_names> ...> [-L <locale>]
```

Parameters

-I <instance_name>

Specifies the name of the InfoSphere CDC instance. Alternatively, you can specify the TSINSTANCE environment variable in place of this value.

-S Specifies that InfoSphere CDC clears events from the source.

-T Specifies that InfoSphere CDC clears events from both the source and target. If none of the S, T, and B options are specified, InfoSphere CDC assumes B by default.

-B Specifies the name of a source table in the subscription on which InfoSphere CDC sets a log position. You must specify the table name in the format *schema.table*. List the tables if you specify more than one.

-A Specifies that InfoSphere CDC clears events for all subscriptions.

-s <subscription_names>

Specifies that InfoSphere CDC clears events for the indicated subscription. List the subscriptions if you specify more than one.

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

Examples

```
dmclearevents -I myinstance -S -A
```

InfoSphere CDC clears events from the source for **all** subscriptions for the specified instance.

```
dmclearevents -I myinstance -T -s Finance Marketing
```

InfoSphere CDC clears events from both the source and target for the **Finance** and **Marketing** subscriptions for the specified instance.

dmgetsubscriptionstatus - Get Subscription Status

Use this command to retrieve information indicating the current state of subscriptions and to send the results to standard output.

Syntax

```
dmgetsubscriptionstatus -I <instance_name> [-p] <-a|-s <subscription_name> ...> [-L <locale>]
```

Parameters

-I <instance_name>

Specifies the name of the InfoSphere CDC instance. Alternatively, you can specify the TSINSTANCE environment variable in place of this value.

-p Specifies that InfoSphere CDC sends state information to standard output.

-A Specifies that InfoSphere CDC retrieves state information for all subscriptions.

-s <subscription_name>

Specifies the name of the subscription for which state information is retrieved. List the subscriptions if you specify more than one.

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns one of the following:

- 0—If the specified subscriptions have a state of **Inactive**.
- 1—If any of the specified subscriptions have a state other than **Inactive**.
- A negative value—If an error has occurred while retrieving status information.

Examples

```
dmgetsubscriptionstatus -I myinstance -p -A
```

InfoSphere CDC retrieves state information for all subscriptions and sends the results to standard output for the specified instance.

dmshoevents - Display InfoSphere CDC Events

Use this command to display InfoSphere CDC events to standard output. You can use this command as an alternative to showing InfoSphere CDC events in the Event Log view in Management Console.

The output of this command shows events in chronological order with the most recent event shown first in the list.

Syntax

```
dmshoevents -I <instance_name> <-a|-s <subscription> ...> [-t <source_ID> ...|-s <subscription> ... -t <source_ID> ...> [-h] [-c max_msg] [-L <locale>]
```

or

```
dmshoevents -I <instance_name> <-a|-s <subscription>|-t <source_ID> ...> [-h] [-c max_msg] [-L <locale>]
```

Parameters

-I <instance_name>

Specifies the name of the InfoSphere CDC instance. Alternatively, you can specify the TSINSTANCE environment variable in place of this value.

-a Specifies that InfoSphere CDC shows events for all subscriptions.

-s <subscription>

Specifies the name of the source subscription for which InfoSphere CDC shows events. List the subscriptions if you specify more than one.

-t <source_ID>

Specifies the source ID for which InfoSphere CDC shows events. List the source IDs if you specify more than one.

-h Specifies that InfoSphere CDC displays a header before the list of events. This option helps you identify each item of information that is displayed for each event.

-c max_msg

Specifies the maximum number of events that InfoSphere CDC displays. If you omit this parameter or you specify a value greater than the total number of events, InfoSphere CDC displays all events for the specified subscriptions and/or source IDs.

- **Minimum Setting**—0. No events are shown.
- **Maximum Setting**—2147483647

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

Examples

```
dmshoevents -I new_instance -s Finance
```

InfoSphere CDC displays all events for the **Finance** subscription for the specified instance.

```
dmshoevents -I myinstance -a -h
```

InfoSphere CDC displays all events for all subscriptions. A header is displayed before the list of events for the specified instance.

```
dmshoevents -I newinstance -s Finance -t Atlanta -s Marketing -h -c 20  
dmshoevents -I myinstance -s Finance Marketing -t Atlanta -h -c 20
```

InfoSphere CDC displays the most recent 20 events for the **Finance** and **Marketing** subscriptions and for the Atlanta source ID. A header is displayed before the list of events for the specified instance.

Sample output

```
EVENTTIME|EVENTSOURCE|ORIGINATOR|EVENTID|SEVERITY|EVENTPROGRAM|EVENTTEXT
```

```
2006-04-21 17:23:08.817|T|ATLANTA|95|Information|class com.datamirror.ts.target.  
publication.c|Transformation Server Communications ending.
```



```
2006-04-21 17:23:08.614|T|ATLANTA|1538|Information|class com.datamirror.ts.target.
publication.c|---Transformation Server for ATLANTA terminating normally.
```

```
2006-04-21 17:23:08.333|T|ATLANTA|1537|Information|class com.datamirror.ts.target.
publication.c|Describe conversation with ATLANTA completed successfully.
```

```
2006-04-21 17:23:07.911|T|ATLANTA|1536|Information|class com.datamirror.ts.target.
publication.c|Describe conversation started by ATLANTA.
```

```
2006-04-21 17:23:07.333|T|ATLANTA|1531|Information|class com.datamirror.ts.target.
publication.c|Communication with ATLANTA successfully started on Data channel.
```

```
2006-04-21 17:23:06.973|T|ATLANTA|1534|Information|class com.datamirror.ts.engine.a
|Code page conversation from the source database's code page 1252 to the target
database's code page Cp1252 for ATLANTA will be performed by the Remote system
```

Fields in each record are separated by vertical bars (|). These fields are identified in the first line of the output. In the *EVENTSOURCE* field, *S* indicates source and *T* indicates target.

Exporting and importing configuration commands

This section contains commands that allow you to export and/or import your InfoSphere CDC global configuration.

dmexportconfiguration - Export InfoSphere CDC Configuration

Use this command to export the configuration details you had set when you had installed an instance of InfoSphere CDC. Configuration details are sent to an XML configuration file. You can use the `dmimportconfiguration` command to import the XML file that you create with this command into another instance of InfoSphere CDC.

Note: This command does not export subscription-specific settings that are configured in Management Console. Subscription-specific settings can be exported to an XML file in Management Console. For more information, see your Management Console documentation.

This command is interactive and will prompt you for your password. You cannot use this command in a script.

Syntax

```
dmexportconfiguration <path_to_configuration_file> [-L <locale>]
```

Parameters

<path_to_configuration_file>

The relative or absolute path to the XML configuration file that you want to export. The relative path is relative to your installation of InfoSphere CDC.

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

Examples

```
dmexportconfiguration c:\configurations\configuration.xml
```

InfoSphere CDC exports the XML file to specified relative path.

dmimportconfiguration - Import InfoSphere CDC Configuration

Use this command to import InfoSphere CDC configuration settings from an XML file you created with the `dmexportconfiguration` command.

Note: You can use this command in a script in coordination with a silent installation to deploy InfoSphere CDC on multiple systems.

Syntax

```
dmimportconfiguration <path_to_configuration_file> [-L <locale>]
```

Parameters

<path_to_configuration_file>

The relative or absolute path to the XML configuration file that you are importing. The relative path is relative to your installation of InfoSphere CDC.

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

Examples

```
dmimportconfiguration c:\configurations\configuration.xml
```

InfoSphere CDC imports the XML configuration file from the specified relative path.

Other commands

This section contains miscellaneous commands that allow you to determine the version of InfoSphere CDC, verify communications, shutdown, and terminate InfoSphere CDC (UNIX servers only), set system parameters, and backup your metadata.

dmbackupmd - Backup Metadata

Use this command to create a backup copy of your metadata database. The backup copy is created in the `instance/<instance_name>/conf/backup` directory in your InfoSphere CDC installation directory. Back up after making changes to your subscription configuration and table status. You can back up your metadata while InfoSphere CDC is running.

Syntax

```
dmbackupmd -I <instance_name> [-L <locale>]
```

Parameters

-I <instance_name>

Specifies the name of the InfoSphere CDC instance. Alternatively, you can specify the TSINSTANCE environment variable in place of this value.

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

dmset - Set InfoSphere CDC System Parameter

Use this command to view or change InfoSphere CDC system parameters. You can also change system parameters in Management Console. For more information, see your Management Console documentation.

Note: You can set any system parameter using this command. However, it will only display system parameters that are set to non-default values.

Syntax

```
dmset -I <instance_name> [<parameter_name>[=<parameter_value>]] [-L <locale>]
```

Parameters

-I <instance_name>

Specifies the name of the InfoSphere CDC instance. Alternatively, you can specify the TSINSTANCE environment variable in place of this value.

<parameter_name>

Specifies the name of the InfoSphere CDC system parameter.

<parameter_value>

Specifies the value that you want to assign to the system parameter.

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

Examples

```
dmset -I myinstance
```

Displays all of the system parameters that are set to non-default values.

```
dmset -I myinstance global_unicode_as_char=false
```

Sets the global_unicode_as_char system parameter to false.

```
dmset -I myinstance global_unicode_as_char
```

Displays the current value of the specified parameter.

```
dmset -I myinstance stop_replication=
```

Deletes the stop_replication system parameter.

dmshowversion - Show InfoSphere CDC Version

Use this command to display the InfoSphere CDC version and build number. Run this command before you contact your IBM representative, so that you can provide the version and build number of InfoSphere CDC that you are running.

Syntax

```
dmshowversion [-L <locale>]
```

Parameters

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

dmshutdown - Shut Down InfoSphere CDC

Use this command to gracefully shut down InfoSphere CDC. You can use this command prior to taking a server or database offline for maintenance purposes or upgrading InfoSphere CDC to the latest version.

Before you run this command, end replication on all subscriptions in Management Console, to ensure that a shut down is completed. For more information, see your Management Console documentation.

If this command cannot completely shut down InfoSphere CDC, use the dmterminate command to force a complete shut down.

Syntax

```
dmshutdown -I <instance_name> [-L <locale>]
```

Parameters

-I <instance_name>

Specifies the name of the InfoSphere CDC instance. Alternatively, you can specify the TSINSTANCE environment variable in place of this value.

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

dmterminate - Terminate InfoSphere CDC Processes

Note: This command is not supported on Windows.

Use this command to terminate all InfoSphere CDC processes, for all instances running on a UNIX or Linux server that you cannot completely shut down with the `dmshutdown` command. InfoSphere CDC terminates only processes that are started by the UNIX account used to run this command.

You can use this command prior to taking a server or database offline for maintenance purposes or upgrading InfoSphere CDC to the latest version.

Use the `dmshutdown` command to gracefully shut down InfoSphere CDC. If `dmshutdown` is unable to completely shut down InfoSphere CDC, then use `dmterminate` to terminate any active InfoSphere CDC processes that still remain after issuing `dmshutdown`.

Syntax

```
dmterminate [-L <locale>]
```

Parameters

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

dmmts32 - Start InfoSphere CDC

Use this command to start InfoSphere CDC 32-bit version.

Syntax

```
dmmts32 -I <instance_name> [-L <locale>]
```

Parameters

-I <instance_name>

Specifies the InfoSphere CDC instance for which you want to start.

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

Examples

```
dmmts32 -I -I myinstance
```

InfoSphere CDC starts for the specified instance.

dmmts64 - Start InfoSphere CDC

Use this command to start InfoSphere CDC 64-bit version.

Syntax

```
dmts64 -I <instance_name> [-L <locale>]
```

Parameters

-I <instance_name>

Specifies the InfoSphere CDC instance for which you want to start.

-L <locale>

The name of the locale used for the InfoSphere CDC instance. The default is your machine's locale.

Result

This command returns a value of 0 if the command was successful and a non-zero value if the command fails.

Examples

```
dmts64 -I myinstance
```

InfoSphere CDC starts for the specified instance.

User exits for InfoSphere CDC

A user exit lets you define a set of actions that InfoSphere CDC can run before or after a database event occurs on a specified table. User exits allow you to customize your environment to meet your business requirements.

After compiling your Java Class or Stored Procedure user exit, you can configure the user exit in Management Console. For more information on configuring user exits, see "Configuring user exits" in your Management Console documentation.

The Javadoc (API) information that is installed with InfoSphere CDC provides detailed class and interface specifications for the Java Class user exits that are available in InfoSphere CDC. For each interface, the supported methods that can be called are identified.

The Javadoc (API) documentation for the user exits is located in the following directory: <system drive>:\<installation directory>\docs\api. To open the help in your browser, click [index.html](#).

Sample user exits are provided with InfoSphere CDC. You can extend or modify these samples to suit your environment.

Stored procedure user exits for table and row level operations

A stored procedure is a program (or procedure) which is physically stored within a database. The advantage of a stored procedure is that when it is run, in response to a user request, it is run directly by the database engine, which usually runs on a separate database server and is generally faster at processing database requests.

After writing and compiling user exit programs, you can specify at which user exit point you want to invoke the user exit (either before or after a row-level or before or after a table-level operation) on the **User Exits** tab of Management Console.

Defining a stored procedure user exit

When defining a stored procedure in InfoSphere CDC, consider the following:

- Overloaded stored procedures are not supported.
- Stored procedures must have at least two parameters, which must be the first two defined, in the following order:
 - `result`—an integer output parameter, used to return any error codes to the event log.
 - `returnMsg`—a character output parameter, used to return error messages to be logged.

Stored procedure user exit database connections

The stored procedure user exit program and InfoSphere CDC use the same shared connection as the default method to connect to the database. This setting ensures that, by default, changes to tables made by InfoSphere CDC are visible to the stored procedure user exit program.

Retrieving data with a stored procedure user exit

You can retrieve data from your source table by passing system parameters to your stored procedure. You can retrieve the following type of data:

- **Retrieve system values (s\$)**—when passed to a stored procedure, the `s$` prefix makes system values available from the source database to your stored procedure. For example, `s$entry` identifies the entry point at which InfoSphere CDC had run the user exit.
- **Retrieve journal control fields (j\$)**—when passed to a stored procedure, the `j$` prefix makes journal control fields available from the source database to your stored procedure. For example, `j$USER` identifies the user ID of the person that made the update on the source table. This is useful if you are using the stored procedure to audit table or row-level operations that have occurred on the source table.
- **Retrieve data values**—depending on the prefix you pass to the stored procedure, you can retrieve data from the source database and make it available to your stored procedure. For example, you can use `b$` to retrieve the before image of the source column

Each of these values can be used as input parameters for the stored procedure user exit that you write. The format used to retrieve data is slightly different depending on the product that you are using:

- For InfoSphere CDC, the format is `<x>${<value>}`

where `<x>` represents the prefix and `<value>` represents the name of the value to be retrieved.

Retrieving system values using the `s$` prefix

This prefix is used to retrieve system values. The table below presents and briefly describes these values.

Prefix and Value	Data Type	Description
s\$entry	NUMBER	<p>Represents the entry point from where the stored procedure was invoked. You can invoke a stored procedure from the following entry points:</p> <ul style="list-style-type: none"> • 1—indicates that InfoSphere CDC has invoked the stored procedure before a table clear (truncate) operation • 2—indicates that InfoSphere CDC has invoked the stored procedure after a table clear (truncate) operation • 3—indicates that InfoSphere CDC has invoked the stored procedure before a row insert operation • 4—indicates that InfoSphere CDC has invoked the stored procedure after a row insert operation • 5—indicates that InfoSphere CDC has invoked the stored procedure before a row update operation • 6—indicates that InfoSphere CDC has invoked the stored procedure after a row update operation • 7—indicates that InfoSphere CDC has invoked the stored procedure before a row delete operation • 8—indicates that InfoSphere CDC has invoked the stored procedure after a row delete operation • 9—indicates that InfoSphere CDC has invoked the stored procedure before a table refresh operation • 10—indicates that InfoSphere CDC has invoked the stored procedure after a table refresh operation

Prefix and Value	Data Type	Description
s\$srcSysId	VARCHAR	Identifies uniquely the location of the source data.
s\$srcTabId	VARCHAR	Represents the name of the source table in the source database that sends replicated data to the target.
s\$tgtTabId	VARCHAR	Represents the name of the target table in the target database that receives replicated data from the source.

Retrieving journal control fields using the j\$ prefix

This prefix is used to retrieve information about the operation that occurred on the source system. You can use **jb\$** with InfoSphere CDC to retrieve the same information.

The available values are listed:

Prefix and Value	Data Type	Description
j\$CCID	VARCHAR	Identifies the transaction with the insert, update, or delete operation.
j\$CODE	VARCHAR	Identifies the type of journal or log entry, either "U" for a refresh operation or "R" for mirroring.
j\$CTRR or j\$CNTRRN	VARCHAR	Identifies the relative record number of the source table that recorded the journal/log entry. Note: CTRR or CNTRRN contains meaningful information when you invoke your stored procedure on the insert entries that make up the refresh.
j\$ENTT or j\$ENTTYP	VARCHAR	Generates journal or log codes that identify the operation type on the source system.
j\$JRN or j\$JOURNAL	VARCHAR	The name of the journal/log where InfoSphere CDC is reading insert, update, or delete operations from.
j\$JOB	VARCHAR	Identifies the name of the job that made the insert, update, or delete on the source system.
j\$MBR or j\$MEMBER	VARCHAR	Identifies the name of the source table or its alias.

Prefix and Value	Data Type	Description
j\$NBR or j\$JOBNO	VARCHAR	Identifies the process ID of the program on the source table that is making the insert, update, or delete operation.
j\$PGM or j\$PROGRAM	VARCHAR	Identifies the name of the program on the source system that made the insert, update or delete operation.
j\$SEQN or j\$SEQNO	VARCHAR	Identifies the sequence number of the insert, update, or delete operation in the journal or log.
j\$SYNM or j\$SYSTEM	VARCHAR	Identifies the host name of the source system.
j\$USER	VARCHAR	Identifies the database user name that made the insert, update, or delete operation on the source.
j\$USPF	VARCHAR	Identifies the operating system user name that made the insert, update, or delete operation on the source.
j\$TSTP or j\$TIMESTAMP	VARCHAR	Identifies the date and time of when the insert, update, or delete operation or refresh was made on the source. In environments that support microsecond precision, the date and time format of this journal control field is YYYY-MM-DD-HH:MM:SS.UUUUUU. Otherwise, InfoSphere CDC sets the microsecond component UUUUUU to zeroes or does not include it at all.

Retrieving data values using b\$, a\$, k\$, and d\$ prefixes

Four prefixes are used to retrieve data:

Prefix	Mode	Description
b\$<source column name>	Input	<p>Used to retrieve the before image of the data in a source column. The before image is the original image from the source table column before any transformation is applied to it.</p> <p>For example, you may have made the following UPDATE to your source table:</p> <pre>UPDATE source_table set MYCOLUMN = 2 where MYCOLUMN = 1;</pre> <p>This will set 2 on all rows where MYCOLUMN was 1 before the execution of this SQL statement.</p> <p>When you define a stored procedure and decide that you want the stored procedure to retrieve the before image of MYCOLUMN, you would specify the following:</p> <pre>b\$MYCOLUMN;</pre> <p>This returns a value of 1.</p>

Prefix	Mode	Description
a\$<source column name>	Input	<p>Used to retrieve the after image of the data in a source column. The after image is the translated data from the source table column. For example, the data that was translated by a derived expression.</p> <p>For example, you may have made the following UPDATE to your source table:</p> <pre>UPDATE source_table set MYCOLUMN = 2 where MYCOLUMN = 1;</pre> <p>This will set 2 on all rows where MYCOLUMN was 1 before the execution of this SQL statement.</p> <p>When you define a stored procedure and decide that you want the stored procedure to retrieve the after image of MYCOLUMN, you would specify the following:</p> <pre>a\$MYCOLUMN;</pre> <p>This returns a value of 2.</p>
k\$<target key column name>	Input	<p>Used to access the target table to find the rows that need to be modified.</p> <p>Note: Key columns are not available for auditing.</p>
d\$<target column name>	Input/Output	<p>Used to retrieve data values after transformation, which will be used to update the table in the target database. Only these values may be modified by the stored procedure.</p>

Example of a stored procedure user exit

The following code snippet is an example of a stored procedure user exit.

Code	Comments
<pre> create or replace procedure PROD.AUDIT_STPROC (result OUT INT, returnMsg OUT CHAR, s\$entry IN NUMBER, s\$srcSysId IN CHAR, s\$srcTabId IN CHAR, s\$tgtTabId IN CHAR, j\$ENTT IN CHAR, a\$IDNO IN NUMBER, a\$PRICE IN NUMBER, a\$DESC IN CHAR, a\$LONGDESC IN CHAR, a\$TRANSDATE IN DATE, d\$IDNO IN NUMBER, d\$PRICE IN NUMBER, d\$DESC IN CHAR, d\$LONGDESC IN CHAR, d\$TRANSDATE IN DATE) </pre>	<p>The parameters you declare and want to pass to your stored procedure must be valid data types.</p> <p>The following parameters are mandatory and must be declared in your stored procedure:</p> <p>result—Returns a value of '0' which indicates the stored procedure was successful or an error which may be an integer.</p> <p>returnMsg—Returns an error message in the Event Log.</p> <p>The following parameters have been declared in this stored procedure:</p> <ul style="list-style-type: none"> • s\$entry—retrieves the entry point at which the stored procedure was called. In this example, InfoSphere CDC calls the user exit at every entry point. • s\$srcSysId—retrieves the location of source data • s\$srcTabId—retrieves the location of source data • s\$srcTabId—retrieves the name of the source table • s\$srcTabId—retrieves the name of the source table • s\$tgtTabId—retrieves the name of the target table • s\$tgtTabId—retrieves the name of the target table • j\$ENTT—retrieves the journal code that indicates the type of operation that took place on the source table • j\$ENTT—retrieves the journal code that indicates the type of operation that took place on the source table • a\$—retrieves the after image of the IDNO, PRICE, DESC, LONGDESC, and TRANSDATE source columns • a\$—retrieves the after image of the IDNO, PRICE, DESC, LONGDESC, and TRANSDATE source columns • d\$—retrieves the transformed data of the IDNO, PRICE, DESC, LONGDESC, and TRANSDATE target columns • d\$—retrieves the transformed data of the IDNO, PRICE, DESC, LONGDESC, and TRANSDATE target columns
<pre> IS ENTRYPOINT VARCHAR(50); BEGIN CASE s\$entry WHEN 16 THEN ENTRYPOINT := 'User Exit program called Before Insert'; WHEN 1048576 THEN ENTRYPOINT := 'User Exit program called After Insert'; WHEN 64 THEN ENTRYPOINT := 'User Exit program called Before Update'; WHEN 4194304 THEN ENTRYPOINT := 'User Exit program called After Update'; END CASE; </pre>	<p>This stored procedure user exit can be invoked from these entry points.</p>

Code	Comments
<pre>insert into PROD.AUDIT_TABLE1 values (s\$entry, s\$srcSysId, s\$srcTabId, s\$tgtTabId, j\$ENTTT, a\$IDNO, a\$PRICE, a\$DESC, a\$LONGDESC, a\$TRANSDATE, d\$IDNO, d\$PRICE, d\$DESC, d\$LONGDESC, d\$TRANSDATE, ENTRYPOINT);</pre>	<p>This stored procedure user exit will INSERT these values into <i>PROD.AUDIT_TABLE1</i>.</p>
<pre>result := 1; returnMsg := 'OK'; END AUDIT_STPROC;</pre>	<p>This stored procedure user exit was successful. Note: If your stored procedure returns a '0', then a message is generated to the event log.</p>

Sample user exits for InfoSphere CDC

InfoSphere CDC provides sample user exits that you can use as is or modify for your working environment:

- **UserExitSample**—subscribes to replication events to retrieve the details of the events which took place.
- **UserExitSample1**—records new rows inserted into a table on the target and stores them in a text file. The user specifies the name of the text file as a parameter.
- **DEUserExitSample**—used in expressions using the %USER column function. It calculates the sum of the user-supplied parameters (in the expression) and returns the sum incremented by 1.
- **SPUserExitSample**—this sample calls a stored procedure with the image coming from the source.
- **PopWindow**—this sample opens a dialog box with notification information. This sample will work with UNIX and Linux installations if you have a GUI emulator such as X-Windows.

To run the sample user exits without modifying them, you must specify the fully qualified path to the compiled user exit in Management Console; for example, `com.datamirror.ts.target.publication.userexit.sample.UserExitSample`.

Compiled sample user exits are located in the `ts.jar` file which is found in `<InfoSphere CDC installation directory>/lib`. Note that the compiled user exits in the `ts.jar` file have a `*.class` extension.

If you want to modify the sample user exits, you must compile the user exit after you make changes to the source code. This section contains instructions for this procedure on the Windows and Linux platforms.

For more information on how to specify Java class or Stored Procedure user exits in Management Console, see your Management Console documentation.

Note: The user exit class must also be in your classpath.

To compile the sample user exits (Windows) Procedure

1. Stop InfoSphere CDC.
2. Unzip the `samples.jar` file into the `lib` folder in your InfoSphere CDC installation folder. Make sure you maintain the folder structure when unzipping the jar file.

After unzipping the jar file, you will have a folder structure like the following:

```
<InfoSphere CDC installation folder>\lib\com\datamirror\ts\target  
\publication\userexit\sample
```

3. Make your changes to the sample user exit.
4. Compile the modified user exit. For example, if you want to compile `UserExitSample.java`, open a command window, navigate to the `lib` folder and issue the following command:

```
javac -classpath ts.jar;. com\datamirror\ts\target\publication\userexit\sample  
\UserExitSample.java
```

If this command runs successfully, there will be no output on your screen.

Note: Your system must have the Java JDK to run this command.

5. After running the command successfully, navigate to the following directory and confirm that you have created a `UserExitSample.class` file:

```
<InfoSphere CDC installation directory>\lib\com\datamirror\ts\target  
\publication\userexit\sample
```

6. Start InfoSphere CDC.
7. The final step to configure the user exit is to specify the fully qualified path to `UserExitSample` in Management Console. For example:

```
com.datamirror.ts.target.publication.userexit.sample.UserExitSample
```

Note: Do not specify the `.class` extension.

What to do next

For more information on how to specify Java class user exits in Management Console, see your Management Console documentation.

Note: If you plan to use the sample user exits in production environments, you will have to test the samples before they are deployed. IBM does not assume responsibility for adverse results caused by modified or customized user exit classes.

To compile the sample user exits (UNIX and Linux) Procedure

1. Stop InfoSphere CDC.
2. Unzip the `samples.jar` file into the `lib` directory in your InfoSphere CDC installation directory. Make sure you maintain the directory structure when unzipping the jar file.

After unzipping the jar file, you will have a directory structure like the following:

```
<InfoSphere CDC installation directory>/lib/com/datamirror/ts/target  
/publication/userexit/sample
```

3. Make your changes to the sample user exit.
4. Compile the modified user exit. For example, if you want to compile `UserExitSample.java`, open a command window, navigate to the `lib` directory and issue the following command:

```
javac -classpath ts.jar:. com/datamirror/ts/target/publication/userexit/sample  
/UserExitSample.java
```

If this command runs successfully, there will be no output on your screen.

Note: Your system must have the Java JDK to run this command.

5. After running the command successfully, navigate to the following directory and confirm that you have created a `UserExitSample.class` file:

```
<InfoSphere CDC installation directory>/lib/com/datamirror/ts/target  
/publication/userexit/sample
```
6. Start InfoSphere CDC.
7. The final step to configure the user exit is to specify the fully qualified path to `UserExitSample` in Management Console. For example:

```
com.datamirror.ts.target.publication.userexit.sample.UserExitSample
```

Note: Do not specify the `.class` extension.

What to do next

For more information on how to specify Java class user exits in Management Console, see your Management Console documentation.

Note: If you plan to use the sample user exits in production environments, you will have to test the samples before they are deployed. IBM does not assume responsibility for adverse results caused by modified or customized user exit classes.

Conflict resolution audit table

When InfoSphere CDC resolves a conflict between the source and target tables, it records information about the resolution in the `TS_CONFAUD` table. The InfoSphere CDC installer creates this table in the target metadata location that is specified during configuration of InfoSphere CDC.

In this section, you will learn:

Structure of the conflict resolution audit table

You can use the `TS_CONFAUD` table to track how conflict resolution affected your target table. For example, you can query the `AFTERIMG` column to see when a change was made to the target table. Then you can review the contents of the `BEFOREIMG` and `AFTERIMG` columns to see the change on the source table that resulted in the data on the target table. This can help in identifying problems in your conflict resolution strategy.

Conflict detection and resolution is configured in Management Console. For more information, see your Management Console documentation.

The structure of the `TS_CONFAUD` table is as follows.

Column	Description
CNFTIME	The date and time on the target when the conflict was detected.
SRCTIME	The time the conflicting data was applied to the source table.
SRCSYSID	The source ID of the subscription.
SRCSHEMA	The schema or library name for the source table.
SRCNAME	The name of the source table.
SRCMEMBER	This field is blank.
TGTSHEMA	The schema or library for the target table.

Column	Description
TGTNAME	The name of the target table.
OPTYPE	The row-level operation on the source that caused the conflict. The value is one of: <ul style="list-style-type: none"> • 1—a row was inserted into the source table. • 2—a row was updated on the source table. • 3—a row was deleted from the source table.
CNFTYPE	The type of conflict that was detected. The value is one of: <ul style="list-style-type: none"> • 1—a row was inserted into the source table. The key for that row already exists in the target table. • 2—a row was updated or deleted on the source table. The key for that row does not exist in the target table. • 3—a row was updated or deleted on the source table. The images of the source and target tables do not match. • 4—an unexpected conflict was detected.
RESMTD	The conflict resolution method that was used. The value is one of: <ul style="list-style-type: none"> • 1—source wins • 2—target wins • 3—largest value wins • 4—smallest value wins • 5—user exit <p>If the resolution method was None, then a row will not be entered into this table. See your InfoSphere CDC documentation for more information on these methods.</p>
CNFRES	Indicates if the conflict was resolved. The value is one of: <ul style="list-style-type: none"> • Y—the conflict was resolved. • N—the conflict was not resolved.
BEFOREIMG	A representation of the row in the source table before it was changed. See “Row image format” on page 78 for more information on the format of this column.
BEFORETRNC	Indicates if the before image stored in BEFOREIMG was truncated. The value is one of: <ul style="list-style-type: none"> • Y—the value was truncated. • N—the value was not truncated.
AFTERIMG	A representation of the row in the source table after it was changed. See “Row image format” on page 78 for more information on the format of this column.
AFTERTRNC	Indicates if the after image stored in AFTERIMG was truncated. The value is one of: <ul style="list-style-type: none"> • Y—the value was truncated. • N—the value was not truncated.
TGTIMG	A representation of the row in the target table before replication occurred. See “Row image format” on page 78 for more information on the format of this column.

Column	Description
TGTRNC	Indicates if the image stored in TGTIMG was truncated. The value is one of: <ul style="list-style-type: none"> • Y—the value was truncated. • N—the value was not truncated.
WINIMG	A representation of the final row in the target table after conflict resolution has occurred. See “Row image format” for more information on the format of this column.
WINTRNC	Indicates if the image stored in WINIMG was truncated. The value is one of: <ul style="list-style-type: none"> • Y—the value was truncated. • N—the value was not truncated.

Row image format

The BEFOREIMG, AFTERIMG, TGTIMG, and WINIMG columns in the audit table show representations of a row in either the source or target table.

The images in these columns are limited by the maximum length of VARCHAR data on your target metadata database. The images contain all of the values in the row, except for data in raw, binary, or LOB columns. The data from each column is presented in the following format:

(length:value)

In the format above, *value* is the data in the column and *length* is the number of characters used to represent the data. The images display numeric data as character strings and NULL values as (null).

The row images match the column order in the source table and the conflict resolution audit table. These images may be truncated if the image is longer than the maximum length of VARCHAR data in the target metadata database. If a table's key column is not the first column in the table, then it may be truncated.

Truncated images

If a row image is longer than the maximum length of a VARCHAR column, then they will be truncated. There is a column in the audit table that indicates if each image column has been truncated. For example, if WINTRNC is Y, then the value of WINIMG was truncated. The format of the truncated column is:

(-length:value)

In the format above, *value* is the truncated value and *length* is the number of characters in the truncated string.

Unaudited data types

The audit table does not include columns of the following data types in its images:

- IMAGE
- NTEXT
- TEXT

If the source or target table contains rows with these data types, then the image simply overlooks them. Binary data will appear in the images as hex-encoded characters. The image does not store any information from unsupported columns.

Appendix to Management Console Administration Guide

This section provides information on the user exits and system parameters that are to specific solidDB. This section is an appendix to the *InfoSphere Change Data Capture Management Console, Administration Guide*.

Configuring user exits

A user exit lets you define a set of actions that Transformation Server® can run before or after a database event occurs on a specified table. When using Transformation Server, a database event is defined as either a row-level operation or as a table-level operation. Row-level operations include an insert, update, or a delete. Table-level operations include a refresh or a truncate operation. For example, you can configure a row-level user exit program that sends an alert after Transformation Server replicates a delete operation on a particular target table.

User exits can be grouped as either a Before User Exit or an After User Exit:

- **Before User Exit**—runs before Transformation Server replicates any row-level or table-level operations to the target table.
- **After User Exit**—runs after Transformation Server replicates any row-level or table-level operations to the target table.

The following list identifies common scenarios for developing a user exit program before or after row or table-level operations:

- Customize when Transformation Server replicates a row-level operation to the target table. For example, you can develop logic for insert, update, or delete operations so that these occur based on some specified criteria, such as the original invoice date. Transformation Server can run the user exit and apply the row-level operation (insert, update, or delete) to the appropriate target table based on the original invoice date, such as, January 2004, February 2004, November 2006, and so on.
- Disable the default row-level or table-level operations, and replace them by invoking a user exit program that performs custom operations. For example, in response to a table-level truncate operation, you can develop a user exit that lets you do a soft delete rather than a hard delete on the target table.

Configuring user exits for Transformation Server for solidDB

With Transformation Server for solidDB, you can configure Java Class user exits.

For Java class user exits, method names are pre-defined. This means that you can only enable and disable user exit programs. You need to configure a user exit in Java that implements the UserExitIF interface class provided by Transformation Server for solidDB.

To configure a user exit for a Java class:

Procedure

1. Click **Configuration > Subscriptions**.
2. Select the subscription.
3. Click the **Table Mappings** view and select the table mapping.
4. Right-click and select **Edit Mapping Details**.
5. Click the **User Exits** tab.
6. Select **Java Class** from the **User Exit Type** list.
7. Type the name of the Java class user exit that implements the **UserExitIF** interface in the **Class Name** box.

For example, you may have imported the UserExitIF interface, and the user exit program class that implements this interface in your function has the following definition: public class UE1 implements UserExitIF

In the **Class Name** box, you need to type:

Option	Description
UE1	If it is a stand-alone class
<Java package>.UE1	If the class is included in a Java package (for example, com.datamirror.interface.UE1)

The files you generate from compiling the user exit program must be located in a library or folder that is referenced by the CLASSPATH environment variable.

8. Type the parameters that you want to make available to the user exit program in the **Parameter** box.

You can access the parameters in the user exit program class by invoking the **getParameter()** method during the initialization process. There are no conventions for specifying the parameters. The values you type in this box are free-form. The string of parameter values cannot exceed 255 characters in length.

9. Type the name of the user exit programs you want Transformation Server to call beside one or more of the following operations:

Option	Description
Before Insert	Transformation Server runs the user exit before replicating an insert operation.
After Insert	Transformation Server runs the user exit after replicating an insert operation.
Before Update	Transformation Server runs the user exit before replicating an update operation.
After Update	Transformation Server runs the user exit after replicating an update operation.
Before Delete	Transformation Server runs the user exit before replicating a delete operation.
After Delete	Transformation Server runs the user exit after replicating a delete operation.
Before Refresh	Transformation Server runs the user exit before replicating a refresh operation.
After Refresh	Transformation Server runs the user exit after replicating a refresh operation.
Before Truncate	Transformation Server runs the user exit before replicating a truncate operation.
After Truncate	Transformation Server runs the user exit after replicating a truncate operation.

10. Click **Apply**.

System parameters for Transformation Server for solidDB

System parameters let you control the behavior of Transformation Server. If your replication environment requires a particular configuration, then you can use system parameters to modify the behavior of default operations in Transformation

Server. The default system parameter settings are appropriate for most installations. Maintain these default settings until you become familiar with the configuration of Transformation Server.

Transformation Server provides system parameters that control the behavior of your source and target datastores.

Notes:

- If you make changes to a system parameter during active replication, you must stop and restart Transformation Server for the changes to take effect.
- When upgrading to a higher version of Transformation Server, any existing settings for system parameters are maintained.

Notification system parameters

Notification system parameters let you control if you should generate Transformation Server messages in the **Event Log** for specific events.

global_shutdown_after_no_heartbeat_response_minutes:

Use this system parameter to specify the duration, in minutes, of communication inactivity before active Transformation Server processes for a subscription are stopped. If a value outside the acceptable range is specified, the default setting is used.

Applies To—Source

Default Setting—15 minutes

Minimum Setting—3 minutes

Maximum Setting—999 minutes

global_conversion_not_possible_warning:

Use this system parameter to control whether or not Transformation Server generates a warning in the Management Console **Event Log** in the following situations:

- Data conversion is not possible for a specific data value.
- Converted data types are encountered that are out of range.

Set this parameter to one of the following:

true—generates a warning in the **Event Log** if data conversion is not possible for a specific data value or converted data types are encountered that are out of range.

false—does not generate a warning in the **Event Log** if data conversion is not possible for a specific data value or converted data types are encountered that are out of range.

Applies To—Target

Default Setting—False

Maximize throughput system parameters

Transformation Server system parameters allow you to significantly reduce the workload of the target database during mirroring. The Transformation Server apply process groups transactions on the target to reduce the workload. Every commit on the target database will correspond with a commit on the source. However, it may not perform every commit that was done on the source. For example, if the source does three small transactions containing one operation each, the target may commit all three operations as part of a single transaction. You can use this grouping of system parameters to significantly reduce the resources required by the target database. The default settings are appropriate for most databases, but if your target system has limited resources and an increase in latency is acceptable, you can adjust the settings appropriately.

mirror_commit_after_max_transactions:

This system parameter specifies the maximum number of transactions that are grouped together before a commit. Normally, commits issued to the target database are in response to commits issued by applications running on the source. You can use this system parameter to manage commits by controlling how often they are issued to the target database. This approach can be used to reduce the overhead of frequent commits to the database.

Applies To—Target

Default Setting—10

Minimum Setting—1

mirror_commit_after_max_seconds:

This system parameter specifies the amount of time, in seconds, before committing small transactions to the target database. Normally, commits issued to the target database are in response to commits issued by applications running on the source. You can use this system parameter to manage commits by controlling how often they are issued to the target database. This approach can be used to reduce the overhead of frequent commits to the database.

Applies To—Target

Default Setting—1 second

Minimum Setting—1

mirror_commit_after_max_operations:

This system parameter specifies the number of operations that must be applied to the target database before a commit is issued. Normally, commits issued to the target database are in response to commits issued by applications running on the source. You can use this system parameter to manage commits by controlling how often they are issued to the target database. This approach can be used to reduce the overhead of frequent commits to the database.

Applies To—Target

Default Setting—1000

Minimum Setting—1

mirror_commit_on_transaction_boundary:

This system parameter indicates whether or not the commits that Transformation Server does on the target database will always correspond with a commit that occurred on the source database. If you choose to ignore the commitment control of the source database, Transformation Server allows you to see the partial results of large transactions.

Set this parameter to one of the following:

- **true**—Does not ignore the commitment control of the source database. Only records in a committed transaction are mirrored to the target. This setting provides true transaction consistency by ensuring that only committed transactions are sent to the target.
- **false**—Ignores the commitment control of the source database. This value disables commitment control for transaction processing. No attempt to maintain transaction consistency is performed during mirroring.

Applies To—Target

Default Setting—true

refresh_commit_after_max_operations:

This system parameter identifies the number of rows comprising each transaction during refresh. To reduce the workload on the target database during refresh, Transformation Server periodically commits the changes to the target database rather than performing the refresh as a single large transaction.

Applies To—Target

Default Setting—1000

Minimum Setting—1

Encoding system parameters

For some system parameters, you can set the default method for treating data in defined Unicode columns, and you can set the default character encoding for your database.

global_unicode_as_char:

This system parameter indicates the default method of treating data in defined Unicode columns. For each Transformation Server installation on a server, this system parameter defines the system default method of treating data in Unicode columns. If a Unicode column is set to the system default, the current system default method, as defined by this system parameter, is used.

Set this parameter to one of the following:

- **true**—Transformation Server treats all data in Unicode columns as single-byte characters. Use this setting when Unicode columns contain single-byte character data.
- **false**—Transformation Server treats all data in Unicode columns as a continuous bit stream. Use this setting when Unicode columns contain non-single-byte

character data. Setting this system parameter to false ensures that Transformation Server handles non-single-byte character data in the same way as previous Transformation Server releases.

Note: Setting this parameter to false does not ensure that replicated non-single-byte character data in Unicode columns are represented properly on the target. For replicated non-single-byte character data, you may have to apply user exit programs or other customization to properly represent data in Unicode columns. For more information about user exit programs, see the *Transformation Server End-User Documentation* for your platform.

Applies To—Source

Default Setting—false

Disk resource system parameters

Some system parameters control memory usage in Transformation Server. For improved performance, if you are able to allocate more than the default value of 512 MB for the Transformation Server Java Virtual Machine, then you can adjust the disk resource system parameters to use the increased memory.

mirror_memory_txqueue_total_mb:

This system parameter controls the amount of memory used to stage data on the source. For optimal performance, this system parameter should be large enough to hold the largest amount of uncommitted data that will ever exist in the source database.

Applies To—Source

Default Setting—15 Megabytes

mirror_memory_txqueue_each_mb:

This system parameter controls the amount of memory used to stage data on the source. For optimal performance, this system parameter should be large enough to hold the data for the largest transactions that occur on the source.

Applies To—Source

Default Setting—3 Megabytes

global_memory_lob_cache_mb:

This system parameter controls the amount of memory that will be used to stage LOB values on the target. For optimal performance, this value should be large enough to hold the entire data for the largest LOB values that will be replicated.

Default Setting—2 Megabytes

Applies To—Target

mirror_queue_for_buffers_between_cdc_threads_operations:

This system parameter controls the ability of Transformation Server's log scraping to take advantage of multiple processors. The default setting is acceptable for most situations. You can increase this value for highly scalable environments.

Applies To—Source

Default Setting—100 entries

Minimum Setting—100 entries

Apply process system parameters

Some system parameters adjust the way Transformation Server applies rows, column data, and error handling.

mirror_end_on_error:

Use this system parameter to indicate if you want to end mirroring after an apply error occurs on the target database.

Set this parameter to one of the following:

- true—End mirroring after an apply error on the target database.
- false—Do not end mirroring after an apply error on the target database.

Applies To—Target

Default Setting—true

refresh_end_on_error:

Use this system parameter to indicate if you want to end a refresh after an apply error occurs.

Set this parameter to one of the following:

- true—End a refresh after an apply error occurs.
- false—Do not end a refresh after an apply error occurs.

Applies To—Target

Default Setting—true

Notices

© Copyright Oy International Business Machines Ab 1993, 2011.

All rights reserved.

No portion of this product may be used in any way except as expressly authorized in writing by Oy International Business Machines Ab.

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 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.



Printed in USA

SC23-9832-03

