Before using this information and the product that it supports, read the information in “Notices” on page 31.
Chapter 1. Introduction

SQL replication consists of the Capture program, the Apply program, the Replication Alert Monitor, Replication Center, the ASNCLP command-line program, and System i® system commands.

Introduction

SQL replication consists of the following major components:

The Capture program

The Capture program reads the source database log or journal to recognize database updates of interest (as specified by the user) and stores information related to these updates in local DB2® tables. Any changes (insert, update, or delete) made to a source table is written to a changed data (CD) table by the Capture program after the transaction is committed. Normally, there is one changed data table for each source table. If the source table is registered in a multiple Capture schema, there are multiple CD tables for that source table. The DB2 commit points for the units of work that include these captured updates are stored in a Capture control table called the unit of work (UOW) table. At a point in time specified by the user, the Capture program deletes data no longer needed in the CD and UOW tables. This Capture delete activity is called pruning.

The Apply program

The Apply program typically runs local to the database that contains the target tables, but can run anywhere in a replication environment as long as it can connect to the source, target, and replication control databases. The Apply program fetches the data stored in the CD tables, stores the fetched rows into one or more spill files, and then applies each change, one row at a time, to the target tables.

The Replication Alert Monitor

The Replication Alert Monitor is a program that allows you to monitor specific activity of the Capture and Apply programs, and of specific subscription sets. You define alert conditions for the Capture and Apply programs, and when the Replication Alert Monitor detects an alert condition, it sends an e-mail notification to a specified contact or contact group. The Replication Alert Monitor runs independently of the Capture and Apply programs and analyzes their execution through information that the Capture and Apply programs store in the replication control tables. Running the Replication Alert Monitor does not affect replication performance.

The Replication Alert Monitor runs in the Linux®, UNIX®, Windows®, and z/OS® operating-system environments. It does not run natively in a System i operating-system environment, but you can monitor System i replication from one of the supported operating systems.

The Replication Center

The Replication Center is a user interface tool that you can use to set up and administer your replication environment and to run the Capture, Apply, and Replication Alert Monitor programs. You can use the Replication Center to perform such administration tasks as:

• Create replication control tables
• Register replication sources
• Create subscription sets and add subscription-set members to the set
• Operate the Capture program
• Operate the Apply program
• Monitor the replication process

The ASNCLP program
The ASNCLP program is a command-line interface for setting up a replication environment. The ASNCLP program is similar to the DB2 CLP, but provides commands specifically for replication. You can use the ASNCLP program to perform such administration tasks as:
• Create replication control tables
• Register replication sources
• Create subscription sets and add subscription-set members to the set

System i system commands
The System i system commands are a command-line interface for setting up a replication environment or System i systems. You can use the System i system commands to perform such administration tasks as:
• Register replication sources
• Create subscription sets and add subscription-set members to the set

SQL replication also includes a number of tools, such as the Replication Analyzer, that help you maintain your replication environment.

When you install DB2, all of these components and tools are available to you. When you install WebSphere® Replication Server for z/OS, you can choose which components you want to use: the Capture program, the Apply program, or the Replication Alert Monitor. When you install DB2 DataPropagator® for iSeries®, the Capture and Apply programs are both installed. The Replication Center runs only on workstation machines, so you do not install it on z/OS or System i.

Although you install the Capture and Apply programs with DB2 or into IBM-supplied libraries (for z/OS and System i), for performance tuning you should treat them as DB2 applications rather than as components of DB2. Thus, the factors that influence the performance of replication are generally the same as what you would expect for any DB2 online transaction processing (OLTP) application: CPU capacity, buffer pool usage, use and number of indexes, statistics, locking, and DB2 system specifications. In addition, some startup and runtime parameters are specific to the Capture and Apply programs.

The performance of a replication configuration is affected by so many different variables that IBM® cannot recommend a single perfect set of values for all customers.
Chapter 2. General tuning advice

You can begin optimizing your SQL replication environment by increasing the performance of your database. After tuning the database, you can tune your replication objects for the best performance.

Linux, UNIX, and Windows database tuning

When you set up a replication environment, be aware that certain DB2 database objects have a significant impact on performance.

Before you focus on tuning replication performance, ensure that your DB2 databases themselves are properly tuned, both for replication and for your application workload. Be sure to tune your source and target databases. When you set up a replication environment, be aware that the following DB2 objects have a significant impact on performance:

- Database objects
- Database parameters
- Database environment variables

Tuning Linux, UNIX, and Windows database objects

Database objects listed here are the most important for replication performance.

The following are the most important database objects for tuning:

**Bufferpools**
For high performance, you should define multiple bufferpools for replication and your application workload. Define as much bufferpool space as you can for optimal performance.

**Disk system**
Whenever possible, use multiple disk drives to allow for parallel I/O. Use disk controllers with fast-write cache capability.

**Logs**
Whenever possible, use multiple disk drives to allow for parallel I/O. For improved performance, use disk striping for the DB2 logs. Store the logs and the data on different sets of disk drives.

**Spill files**
The file system (the value of the `apply_path` startup parameter) for the Apply spill files should span across multiple disk drives to allow for parallel I/O. If possible, store the spill files on a set of disk drives that is separate from the data and logs.

**Statistics**
Ensure that database statistics have been set to allow the DB2 optimizer to use indexes for the replication control tables by running the RUNSTATS utility for all tables. For the CD tables and UOW table, run the RUNSTATS utility just once, when the CD table contains a large amount of data (that
is, when the cardinality is high enough to guarantee that the DB2 query optimizer will use the CD-table index for queries, set aside sufficient space for sorts, if needed, from the sort heap, and create an optimal execution plan. Although the CD tables and UOW table are created with the VOLATILE keyword, statistics that reflect high cardinality are more likely to produce an optimal execution plan.

Save a good set of “mimic stats” information for possible future use by issuing the following command:

db2look -d dbname -a -m -t tablename -o tablename.stat

Table spaces
Tables that have extensive data changes (for example, CD tables) should use DMS raw-device table spaces for optimal performance. Whenever possible, define table space containers that span multiple disk drives to allow for parallel I/O.

**Tuning Linux, UNIX, and Windows database parameter settings**

Some database parameters have a significant impact on replication performance.

You can use the recommended parameter values that are listed in this section to optimize replication tuning for Linux, UNIX and Windows.

The majority of activity for SQL replication consists of SQL INSERT, UPDATE, and DELETE statements. From a DB2 perspective, this activity makes extensive use of locking, logging, and writing of change data from bufferpools to disks. Therefore, IBM recommends that you set the DB2 database parameters listed in Table 1. The values given reflect reasonable performance for running the Capture and Apply programs with automatic pruning of the CD and UOW tables enabled. The values given do not reflect your particular application workload or other DB2 workload. You might need to adjust the values shown in Table 1 to accommodate this extra workload. Any database parameter that is not listed in Table 1 can remain set at the DB2-shipped default value, unless your workload requires a different value.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Recommended value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBHEAP</td>
<td>2048 pages</td>
<td>This parameter is automatically set by default. If you do not want to set the value to automatic, set a large size for the database heap to accommodate a large log buffer size. Increase this value if you have a large number of tables, indexes, table spaces, and buffer pools.</td>
</tr>
<tr>
<td>INTRA_PARALLEL</td>
<td>NO</td>
<td>Because SQL replication uses relatively simple SQL statements, it does not require intra-partition parallelism for the source, target, or replication control server databases. If your applications require intra-partition parallelism, you can set this parameter to YES.</td>
</tr>
<tr>
<td>LOCKLIST</td>
<td>2048 pages</td>
<td>SQL replication locks tables and rows while writing to the CD tables, replication control tables, and the target tables. This parameter is automatically set by default. You can manually set this parameter if you do not want to use the automatic settings. Increase this value if you have many long-running transactions or if you run multiple instances of the Capture program for a single database.</td>
</tr>
<tr>
<td>Parameter</td>
<td>Recommended value</td>
<td>Explanation</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>LOGBUFSZ</td>
<td>512 pages</td>
<td>To allow the Capture program to read log records from memory (log buffers) rather than from disk as often as possible, set aside a large amount of the database heap as a buffer for log records. Remember that this parameter must be less than or equal to the DBHEAP parameter; do not set this value to more than 35 percent of the database heap size. Unless your environment requires a higher value for this parameter, do not set the log buffer size to more than 512 pages because larger log buffers can affect source-transaction response times.</td>
</tr>
<tr>
<td>LOGFILSIZ</td>
<td>8000 pages</td>
<td>To reduce the frequency of log archiving, set aside a large amount of space for the log file. However, large values can increase database-recovery time.</td>
</tr>
<tr>
<td>LOGPRIMARY</td>
<td>60</td>
<td>Because SQL replication increases the amount of logging required for each database, increase the number of primary log files to be preallocated. By preallocating a larger number of primary log files, you can reduce the overhead of allocating new logs during log archiving. The value of 60 for this parameter is an average value that is appropriate for most replication workloads. If your environment requires more (or fewer) preallocated primary log files, you can set this value appropriately, along with LOGFILSIZ and LOGSECOND.</td>
</tr>
<tr>
<td>LOGSECOND</td>
<td>20</td>
<td>Because replication activity can be sporadic, especially for target databases, set aside enough secondary log files to ensure sufficient time for log archiving to complete when the primary log files become full. The value of 20 for this parameter is an average value that is appropriate for most replication workloads. If your environment requires more (or fewer) preallocated secondary log files, you can set this value appropriately, along with LOGFILSIZ and LOGPRIMARY.</td>
</tr>
<tr>
<td>MAXLOCKS</td>
<td>60 percent</td>
<td>To reduce the likelihood of lock escalation (the process of replacing row locks with table locks), increase the percentage of the lock list that must be filled before DB2 performs escalation. Because lock escalation also occurs if the lock list runs out of space, ensure that the LOCKLIST value is high enough for both replication and your application workload.</td>
</tr>
<tr>
<td>NUM_IOCLEANERS</td>
<td>10</td>
<td>Because the transactions for SQL replication primarily change data in the buffer pools, increase the number of asynchronous page cleaners for the database until a high percentage of the database writes are performed asynchronously.</td>
</tr>
<tr>
<td>NUM_IOSERVERS</td>
<td>10</td>
<td>Set this value to the number of processors for the database server to allow for prefetch I/O and asynchronous I/O, but do not set this value lower than the default of 3.</td>
</tr>
<tr>
<td>PCKCACHESZ</td>
<td>2048 pages</td>
<td>In DB2 Version 9, this parameter is automatically set by default. Many of the same SQL statements are used multiple times during replication, a larger package cache size helps DB2 eliminate the need to access the system catalogs when reloading a static packages and to help DB2 eliminate the need for compilation of dynamic SQL.</td>
</tr>
</tbody>
</table>
Tuning UNIX and Windows database environment-variable settings

Set and adjust the DB2 environment-variable settings listed in this section for UNIX and Windows replication.

The values given do not reflect your particular application workload or other DB2 workload. You might need to adjust the values shown in Table 2 to accommodate this extra workload. Any environment variable not listed in Table 2 can remain set at the DB2-shipped default value, unless your workload requires a different value.

Table 2. DB2 environment variables for replication

<table>
<thead>
<tr>
<th>Variable</th>
<th>Recommended value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DB2_MINIMIZE_LIST_PREFETCH</td>
<td>YES</td>
<td>A list-prefetch plan requires a sort, and the sort might lead to an overflow sort which can adversely affect performance. A value of YES ensures that the optimizer will use an execution plan other than a list-prefetch plan, so that sorts can be avoided. However, if your applications require list prefetch, you can set this parameter to NO.</td>
</tr>
<tr>
<td>DB2_MMAP_READ</td>
<td>OFF</td>
<td>Allow DB2 to read from JFS file system cache into memory (outside the buffer pool).</td>
</tr>
<tr>
<td>DB2_PARALLEL_IO</td>
<td>*</td>
<td>Enable parallel I/O for all (*) table spaces.</td>
</tr>
</tbody>
</table>

Database tuning for z/OS and USS

When you set up a replication environment, be aware that certain DB2 database objects and their definitions can have a significant impact on replication performance.

Before you focus on tuning replication performance, ensure that your DB2 databases are properly tuned, both for replication and for your application workload. Be sure to tune your source and target databases. When you set up a replication environment, be aware that the following DB2 database objects have a significant impact on performance:

- Database objects
- Database parameters

Tuning z/OS and USS database objects

The disk drive and memory settings can significantly affect the performance of the z/OS database.

The most important z/OS and USS database objects involved in replication performance when you set up a replication environment are as follows:

Bufferpools

For high performance, you should define multiple bufferpools for replication and your application workload. Define as much bufferpool space as you can for optimal performance.
Disk system
Whenever possible, use multiple disk drives to allow for parallel I/O. In addition, use disk controllers with fast-write cache capability.

Logs
Whenever possible, use multiple disk drives to allow for parallel I/O. For improved performance, use disk striping for the DB2 logs. In addition, store the logs and the data on different sets of disk drives.

Spill files
Use memory for the Capture and Apply spill files. If the Capture and Apply programs run out of memory, they will use Virtual I/O (VIO).

For the USS environment, you cannot specify the CAPSPILL DD statement, so the Capture program uses VIO defined with 50 primary and 100 secondary cylinders. This default size might not be large enough for some transactions, which can cause the Capture program to shut down. You can start the Capture program with JCL to specify a larger CAPSPILL file.

Statistics
Ensure that database statistics have been set to allow the DB2 optimizer to use indexes for the replication control tables by running the RUNSTATS utility for all tables. For the CD tables and UOW table, run the RUNSTATS utility just once, when the CD table contains a large amount of data (that is, when the cardinality is high enough to guarantee that the DB2 query optimizer will use the CD-table index for queries, set aside sufficient space for sorts, if needed, from the sort heap, and create an optimal execution plan).

Tuning z/OS and USS database parameter settings

IBM recommends that you set the DB2 database parameters listed below to optimize replication tuning for z/OS and USS.

The values given reflect reasonable performance for running the Capture and Apply programs with automatic pruning of the CD and UOW tables enabled. The values given do not reflect your particular application workload or other DB2 workload. You might need to adjust the values shown in Table 3 to accommodate this extra workload. Any database parameter not listed in Table 3 can remain set at the DB2-shipped default value, unless your workload requires a different value.

Table 3. DB2 database parameter (DSNZPARM) settings for replication

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Recommended value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLKSIZE</td>
<td>The largest possible block size for the device</td>
<td>Specify the maximum block size of the archive log data set for the device.</td>
</tr>
<tr>
<td>CACHEDYN</td>
<td>YES</td>
<td>Many of the same SQL statements are used multiple times during replication, so allow DB2 to cache prepared, dynamic SQL statements for later use. These prepared statements will be cached in the environmental descriptor manager (EDM) pool. If you specify YES, you should consider this usage when you calculate your EDM pool size.</td>
</tr>
</tbody>
</table>
Table 3. DB2 database parameter (DSNZPARM) settings for replication (continued)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Recommended value</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEALLCT</td>
<td>NOLIMIT</td>
<td>Specifying NOLIMIT allows maximum optimization opportunities. In a data-sharing environment, if the Capture program is stopped for a period of time and all logs are archived to tape, this parameter allows the Capture program to merge the logs without needing to reallocate an archive read tape unit. These units might remain unused for a length of time greater than a specific DEALLCT value. When all tape reading is complete, you can update this option with the SET ARCHIVE command.</td>
</tr>
<tr>
<td>MAXRTU</td>
<td>the number of data-sharing members + 1</td>
<td>This parameter specifies the maximum number of dedicated tape units that can be allocated to read archive log tape volumes concurrently. In a data-sharing environment, if the Capture program is stopped for a period of time and all logs are archived to tape, this parameter allows the Capture program to merge the archived logs and catch up in its processing.</td>
</tr>
<tr>
<td>OUTBUFF</td>
<td>4000 KB or larger</td>
<td>To allow the Capture program to read log records from memory (log buffers) rather than from disk as often as possible, increase the buffer size.</td>
</tr>
<tr>
<td>UNIT</td>
<td>Device type or unit name</td>
<td>Specify the device type or unit name for storing archive log data sets. <strong>Recommendation:</strong> Consider archiving DB2 logs to DASD rather than to tape if the Capture program will often need to capture data from archived logs. If you run the Capture program continuously while DB2 is running, the Capture program will very likely not need to read archive logs. The Capture program is only likely to read archive logs if it has been shut down for a significant period of time while the DB2 system continues to run.</td>
</tr>
</tbody>
</table>

General replication tuning

When you tune your replication environment, you should have a specific goal. This section considers two major goals: maximizing throughput and minimizing processor usage.

Although these goals appear to oppose one another, a number of tuning considerations are common to both. This information unit describes only general tuning advice that can benefit any replication environment.

Among the first tasks that you perform when you set up a replication environment is to register source tables for replication and create subscription sets. After you start the Capture program, it reads the DB2 log or journal and writes changed source rows to a CD table. Then you start the Apply program, which fetches the rows from the CD table to apply the changes to the target tables. To improve overall replication performance, be sure that you replicate only the data that you need by adjusting the registration or subscription set. For example, consider defining triggers on the CD table to reject any inserts of data that you do not want to replicate, such as certain key values or any deleted data, or consider adding predicates to subscription-set members to control how much data is applied to the target tables.

CD tables and performance tuning

The CD tables and UOW table are used by both the Capture and Apply programs. Therefore, these tables have an significant impact on replication performance.
Use the following recommendations concerning CD tables to optimize replication performance:

- When your CD tables contain a large amount of data, run the RUNSTATS utility for each one. You need to collect statistics only once per CD table because the purpose of these statistics is tell the DB2 optimizer to use the associated index for the CD table for queries. There is no equivalent to the RUNSTATS utility for System i, so collecting statistics is not necessary.

- Reorganize the CD tables regularly by using the REORG utility (the RGZPFM command for System i). The CD tables are volatile, and reorganizing them allows DB2 to reuse disk space to use existing space more efficiently.

  Use the REORG utility with the PREFORMAT option to speed up insert processing by the Capture program. For example, DB2 for z/OS adds extents to the end of a data set (which can be expensive), rather than reusing or reclaiming space that is no longer needed in the primary allocation because of Capture pruning.

- You can automatically reorganize the CD tables and UOW table when the Capture job ends by specifying the RGZCTLTBL keyword on the ENDDPRCAP command. Pruning rows from tables removes the data but does not reclaim the disk space until you reorganize the table.

- When you use the Replication Center, asnclp commands, or System i commands to create replication control tables or register a replication source, the generated SQL scripts define specific indexes for each table. In general, do not create extra indexes for CD tables or the UOW table because the generated index is optimal for replication and additional indexes add unnecessary query overhead for the Capture and Apply programs. In certain situations, it can be useful to drop and recreate these indexes with additional columns used by subscription-set member predicates.

- The generated SQL scripts for replication control tables and CD tables also specify a lock size for each table. Do not change these lock sizes, especially for CD tables and the UOW table, from the IBM-supplied values. Lowering lock sizes can degrade performance. Increasing lock sizes can cause contention between the Capture worker and pruning threads.

**Subscription set preparation when tuning for replication**

The Apply program does not need to join the CD and UOW table to populate user copy target tables in most circumstances. When you create a subscription-set member, consider using a user-copy table instead of a point-in-time table or aggregate table whenever possible.

Also, consider removing extra predicates from the subscription set (the UOW_CD_PREDICATES column of the IBMSNAP_SUBS_MEMBR table) that require the Apply program to join a CD table with the UOW table.

If you do add predicates to the subscription set, consider modifying the index for the associated CD table to include the predicate columns so that the Apply program can use the index for its fetches.

**Update-anywhere replication and performance tuning**

If you set up an update-anywhere replication environment, consider designing your source applications so that you can specify “no conflict detection”.

Conflict detection can add to replication latency and increase CPU usage because every changed row must be tested to discover whether there is a conflict. If your
environment has many conflicts, this testing is necessary, but if you have few or no conflicts, this testing can add unnecessary overhead to your environment.

Remote journals

By using remote journaling with replication, you move the CD tables, the Capture program, and the Capture control tables from the System i source server to another System i database server because the source journal entries are written to this other database server.

For optimal performance, this other System i server should also be the replication target server because in this environment, the Apply program can apply changes to the target table without using spill files. Using remote journals for replication has the following benefits:

- More efficient data movement
  If you do not use remote journals, the Apply program moves data from one system to another by using the DRDA® protocol. By using remote journals, the operating system moves the data using a very efficient protocol, and the Apply program does not have to move the data from the source system.

- Less disk space required for the source system
  By using remote journals, the CD tables do not reside on the source system, which can save significant disk space if the transaction volume is high.

- No spill file needed for the Apply job
  If you replicate between System i machines, using remote journals allows the Apply job to reduce its processing time and CPU usage by not having to write to and read from spill files when performing differential refresh.

- Only one license for DB2 DataPropagator for iSeries is required
  If your target server is also the Capture control server (the system with the remote journals), you need only one license for the DB2 DataPropagator product. A license is not needed for the source server.

Restriction: You cannot use remote journals for replication if you replicate LOB columns, or if your replication environment is update-anywhere.

Running the Apply program in a pull configuration and performance tuning

For optimal performance, you will generally run the Apply program at the target server, but you can run it at any DB2 server in your network that can connect to the source server, Capture control server, Apply control server, and target server. When you run the Apply program at the target server, you have a pull configuration: the Apply program pulls updates from the source server to the target server.

Be sure to run the Apply program at the appropriate server for best performance:

- For read-only target tables, the pull configuration enables the best performance, as much as thirty-times better performance compared to a push configuration.

- For update-anywhere replication, run the Apply program on the server where it can pull the most data. For example, if most of your changes are from replica to master, run the Apply program at the master database so that it can pull changes from the replica database.
For peer-to-peer replication, run one Apply program for each peer database. In this case, ensure that you run each Apply program so that it pulls changes from the other peers.

**Replication and performance tuning in a z/OS data-sharing environment**

There are certain recommendations for DB2 when tuning for z/OS and USS in a data-sharing environment.

All of the database tuning advice for DB2 in "Database tuning for z/OS and USS" on page 6, "Tuning z/OS and USS database objects" on page 6, and "Tuning z/OS and USS database parameter settings" on page 7 applies to a data-sharing environment. There are two further recommendations for DB2 when tuning for z/OS and USS in a data-sharing environment.

- Consider running multiple instances of the Capture program for the data-sharing group.
  - To maximize throughput for the Capture program, run each Capture instance on its own member, or on a lightly loaded member. Be sure that each Capture instance uses its own Capture schema. This configuration has dramatically better throughput performance compared to a configuration where a single Capture program runs on the member that generates the most changed data to be captured.

- Use static logical unit (LU) names for the Apply program, whenever possible.
  - A remotely connected Apply program should not use the generic LU capability of DB2 data sharing, but rather should specifically connect to a static LU for the member where the Capture program is running. By connecting directly to the member, the Apply program avoids unnecessary data-sharing global bufferpool overhead.

**Replication trace**

Do not turn the replication trace on unless you need to use it. If the replication trace has been left on unnecessarily, use the asntrc off command to turn it off.

To turn the replication trace off in System i environments, use the WRKDPTRTRC OPTION(*OFF) command or for the Apply trace, specify TRACE(*NONE) for the STRDPRAPY command.

**Startup parameters for the Capture and Apply programs**

Certain startup parameters for the Capture and Apply programs affect replication performance. Use the following recommendations to maximize replication throughput and minimize CPU usage.

Each of the startup parameters has a corresponding value in the IBMSNAP_CAPPARMS or IBMSNAP_APPPARMS table, although those values are not listed here. All startup parameters for the Capture program apply to both the asncap and STRDPRCAP commands, and all startup parameters for the Apply program apply to both the asnapply and STRDPRAPY commands, unless otherwise stated.

**autoprune (for asncap command)**

Set `autoprune` to `y`. 
This parameter specifies whether or not the Capture program automatically prunes some of its control tables. Because the Capture program has a separate pruning thread, there is no need to perform manual pruning during off-peak hours. Be sure to set the pruning interval to an appropriate value for your environment.

**CLNUPITV (for STRDPRCAP command)**

Set `CLNUPITV` to `IMMED` or `DELAYED`.

This parameter specifies the maximum amount of time (in hours) before the Capture program prunes old records from the CD, IBMSNAP_UOW, IBMSNAP_SIGNAL, IBMSNAP_CAPMON, IBMSNAP_CAPTRACE, and IBMSNAP_AUTHTKN tables. Set this parameter so that the Capture program prunes old records at the start or end of the specified interval, and at each interval thereafter.

**memory_limit (for asncap command) or MEMLMT (for STRDPRCAP command)**

Set `memory_limit` or `MEMLMT` to the default or higher value.

This parameter specifies the maximum size (in megabytes) of memory that the Capture program can use to build transactions. After reaching this memory limit, the Capture program spills transactions to a file (or to VIO for z/OS). By setting this parameter to a high value, you reduce the overhead for disk I/O, which improves throughput and saves extra CPU processing.

The following three considerations affect how much memory the Capture program requires:

- The number of concurrent transactions.
- The size of each transaction.
- The value of the Capture `commit_interval` or `FRCFRQ` keyword.

Memory is not released for captured transactions until the Capture program has successfully inserted them into the CD tables and committed the changes to DB2, which occurs at each `commit_interval`.

When running in `asynchlogrd` mode, spilling usually occurs less frequently. In this mode, two threads process log records: one reads the log, and the second writes the captured transactions to the CD tables. The log reader does not spill if the reader has transactions to insert into the CD tables. If memory is about to fill, and there is at least one more transaction to be published, then the log reader sleeps before spilling, up to the time specified by `sleep_interval`, but not longer than the `commit_interval` time.

Use the statistics in the Monitor table to determine if you need a higher value for the `memory_limit` keyword. See [Chapter 5, “Measuring replication performance,” on page 23](#) for more information.

**z/OS** The `memory_limit` value should not exceed 80% of the value for the `REGION` keyword. You can specify 0 so that the Capture program determines the proper value.

**opt4one (for asnapply command) or OPTSNGSET (for STRDRPAPY command)**

If possible, set `opt4one` to `y` or `OPTSNGSET` to (*YES*).

This parameter specifies whether Apply program processing is optimized for a single subscription set. Whenever possible, consider combining all of your source-target pairs into a single subscription set (for a particular Apply qualifier) and setting this parameter to `y` or (*YES*). The Apply program caches and reuses the information about the subscription-set members, which reduces CPU usage and improves throughput rates.
Chapter 3. Tuning replication to maximize throughput

To maximize replication throughput, changes must be written to and then fetched from the changed-data (CD) tables quickly and as often as possible.

To maximize replication throughput, SQL replication must perform both of the following tasks:
- Capture changes from the DB2 log or journal and write these changes to the CD tables as quickly and as often as possible.
- Fetch the changes from the CD tables and apply them to the target tables as quickly and as often as possible.

When you tune replication for maximum throughput, you must often increase the resource consumption for both the source and target databases. This section assumes that your system and database resource needs are secondary to throughput.

To improve the replication throughput for both the Capture and Apply programs, consider creating a set of dedicated bufferpools for the CD tables and the UOW table to improve the Apply fetch rate and reduce I/O for both the Capture and Apply programs.

Tuning the Capture program to maximize throughput

Maximizing the throughput of the Capture program involves using multiple schemas and multiple partitioned tables, adjusting the job or process priority, and adjusting the startup parameters of the Capture program.

Multiple Capture schemas when tuning the Capture program to maximize throughput

In an environment where you can divide your workload to use multiple instances of the Capture program, you can greatly increase your overall throughput.

Using two Capture instances, you might see as much as a fifty-percent increase in overall throughput compared with using just one instance. When using three instances, you can expect a slightly higher increase, but running more than three instances of the Capture program for the same DB2 log or journal does not generally continue to increase overall throughput.

Multiple Capture schemas enable you to run multiple Capture programs concurrently. You might want to run multiple Capture programs in the following situations:
- To optimize performance by treating low-latency tables differently from other tables. If you have low latency tables, you might want to replicate those tables with their own Capture program. That way, you can give them a different runtime priority. Also, you can set the Capture program parameters, such as pruning interval and monitor interval, to suit the low latency of these tables.
- To potentially provide higher Capture throughput. In a source environment with multiple processors, the benefit can be significant. The trade-off for the higher throughput is additional CPU overhead associated with multiple log readers.
You can use multiple Capture schemas if you want to work with Unicode and EBCDIC encoding schemes separately, or if you want to run more than one instance of the Capture program on a subsystem.

**Multiple partitioned tables when tuning the Capture program to maximize throughput**

You can capture changes to source tables that are spread across multiple partitions. For optimal performance, place all control and CD tables in the same database partition group in which the Capture program is running.

It is also recommended that DB2 be tuned so that capturing changes on multiple partitioned tables can achieve top performance levels.

**Job or process priority when tuning the Capture program to maximize throughput**

Although raising the process priority can increase Capture program throughput, it could also decrease overall transaction throughput for a source database, so set the priority appropriately.

- **System i** Consider increasing the priority of the Capture journal jobs if you do not use remote journals.

- **UNIX** In a C program on UNIX systems, you can use the `setpri(process_id, priority)` command to set the process priority.

- **z/OS** Use the Workload Manager (WLM) to set the process priority.

- **System i** Use the `WRKACTJOB` command to select a particular Capture job. On the Work with Active Jobs panel, enter a 2 to change that job. On the Change Job panel, enter a new value in the **Run Priority** field to change the priority for the job.

**Capture program startup parameters important when tuning to maximize throughput**

Certain startup parameters for the Capture program affect replication performance. Use the following recommendations to maximize replication throughput.

**asynchlogrd (for asncap command)**

Set `asynchlogrd=y` to prompt the Capture program to use a dedicated thread for capturing transactions from the DB2 recovery log. The transaction reader thread prefetches committed transactions in a memory buffer, from which another thread gets the transactions and processes them into SQL statements for insertion into the CD table. This asynchronous mode can improve Capture performance in all environments with particular benefits for partitioned databases and z/OS data-sharing. On systems with very high activity levels, this prefetching might lead to more memory usage. Adjust the **memory_limit** parameter accordingly.

**commit_interval (for asncap command) or FRCFREQ (STRDPRCAP command)**

Set `commit_interval` or `FRCFREQ` to a low value.

- **z/OS** This parameter specifies the approximate number of seconds that will elapse between Capture commit
points. The smaller the value that you specify, the sooner changed data will be available to the Apply program, which fetches only data that has been committed by the Capture program. A low value also increases the overall I/O volume for the source system because the Capture program commits its data more frequently. How low to set this value depends on whether dynamic SQL caching is available. If it is available, set this parameter to a low value, such as 1 second. If you do not use dynamic SQL caching, then set this parameter to a higher value, such as 10 seconds.

**System i**
This parameter tells Capture how often to write rows to the CD tables, which is especially important for programs that do not use commit control. The smaller the value that you specify, the sooner changed data will be available to the Apply program, which fetches only data that has been committed. Set this parameter to the minimum value, 30 seconds.

**monitor_interval (for asncap command) or MONITV (for STRDPRCAP command)**
Set **monitor_interval** to a value that fits the needs of your environment.

This parameter specifies how frequently (in seconds) the Capture program inserts rows into the Capture monitor (IBMSNAP_CAPMON) table. Setting a high value reduces overhead, but the accuracy and timeliness of the data can be incomplete. For example, if you are using the alert monitor program to monitor capture latency, then the capture monitor table data must be written frequently enough to support an accurate latency check.

**System i**
An interval of less than 120 seconds is not supported in a System i environment. Slow performance could result if the interval is too small.

**sleep_interval (for asncap command)**
Set **sleep_interval** to 1 second.

**z/OS**
For z/OS non-data-sharing environments and Linux, UNIX, and Windows environments, this parameter specifies the number of seconds that the Capture program sleeps when it finishes processing the active log and determines that the buffer is empty. Set this parameter to a low value for low-latency environments to ensure that the Capture program is always ready to read the log when changes occur. IBM does not recommend that you set this value to 0 because the Capture program will continuously read the DB2 log, which can increase both I/O volume and CPU usage for the Capture server.

**z/OS**
For z/OS data-sharing environments, this parameter specifies the number of seconds that the Capture program sleeps after the buffer returns less than half full.

**WAIT (for STRDPRCAP command)**
Set **WAIT** to a high value.

**System i**
This parameter specifies how the Capture control job responds to external events. A higher value increases the time that the Capture program takes before ending or initializing, but can improve system performance. A value that is too high can result in decreased responsiveness while the Capture program is performing periodic processing. The amount of the decrease in responsiveness depends on the amount of change activity to source tables and the amount of other work occurring on the system.
Tuning the Apply program to maximize throughput

Maximizing the throughput of the Apply program involves adjusting subscription sets, the job or process priority, and the startup parameters of the Capture program. You can also run multiple instances of the Apply program and reduce the number of indexes on source tables to increase throughput.

Subscription set definition when tuning the Apply program to maximize throughput

Define the frequency of replication for the subscription set as continuous to allow the Apply program to process the subscription set as often as possible.

In the Replication Center, on the Schedule page of the Create Subscription Set notebook, select the Time-based check box and select Continuously.

Group multiple members into one subscription set whenever possible. By grouping members into a single subscription set, you reduce the number of database connections for the Apply program to process the subscription-set members and you reduce the administration overhead for maintaining your replication environment.

Consider defining one subscription set that includes all of your source-target pairs and use the Apply startup parameter opt4one (for the asnapply command) or OPTSNGSET (for the STRDRPAPY command), which loads the control-table information for the subscription set into memory. When you specify this parameter, the Apply program does not read the control tables for the subscription-set information for every Apply cycle.

When you create a subscription set that is likely to produce large answer sets, specify the number of minutes’ worth of data for the Apply program to retrieve from the replication source during each Apply cycle. The number of minutes that you specify is called the data block, and the practice, data blocking. If the accumulation of data is greater than the size of the data block, then the Apply program converts a single Apply cycle into several mini-cycles. For target tables that are updated throughout the day by the Apply program, and that are also read by other processes, keeping the Apply answer sets small might allow row-level locking. Row-level locking can reduce contention and improve latency. By retrieving smaller sets of data, the Apply program can lessen both the network load and the temporary space required for the retrieved data.

To set the size of the data block using the Replication Center, set the Data blocking factor on the Create Subscription Set window or use the MAXSYNCH parameter for the System i ADDDPRSUB command.

External CCD targets

You can improve performance for external CCD targets by specifying the type 9 CCD table. This target type allows the Apply program to bypass a join of the change-data (CD) table and the IBMSNAP_UOW table to obtain transaction information. The Apply program obtains the source commit timestamp for the IBMSNAP_LOGMARKER column in the CCD table by generating the same value in the IBMSNAP_LOGMARKER column for all rows in the same cycle.

To specify a type 9 CCD table with the ASNCLP command-line program, use the CREATE MEMBER command and specify NO JOIN CD UOW. In the Replication
Center, use the CCD Properties tab of the Member Properties notebook, and
deselect the Join the CD and UOW table checkbox.

**Remote journal use when tuning the Apply program to maximize throughput**

For System i environments, if you use remote journals for replication, run the
Apply program on the same system as the Capture server. In this environment, the
Apply program can apply changes to the target table without using spill files.

See "Remote journals" on page 10 for more recommendations for using remote
journals with replication.

**Restriction:** You cannot use remote journals for replication if you replicate LOB
columns, or if your replication environment is update-anywhere.

**Running the Apply program when tuning to maximize throughput**

Run multiple instances of the Apply program to increase overall replication
bandwidth. Each Apply instance uses a separate Apply qualifier and processes a
unique group of subscription sets.

Run the Apply program in a pull configuration, as described in "General
replication tuning" on page 8.

Define multiple files for the Apply spill files to allow the Apply program to fetch
large answer sets efficiently.

**Job or process priority when tuning the Apply program to maximize throughput**

Raise the job or process priority for the Apply program.

In a C program on UNIX systems, you can use the
`setpri(process_id, priority)` command to set the process priority.

Use the Workload Manager (WLM) to set the process priority.

Use the WRKACTJOB command to select a particular Apply
job. On the Work with Active Jobs panel, enter a 2 to change that job. On the
Change Job panel, enter a new value in the Run Priority field to change the
priority for the job.

**Indexes for target tables when tuning the Apply program to maximize throughput**

When you use the Replication Center asnclp command, or System i commands to
create the target tables (as part of the task of creating subscription sets), the
generated SQL scripts define a specific index for each table. This index is
optimized for Apply processing.
Consider whether you need extra indexes for the target tables, because with fewer indexes the Apply program can insert and delete data more efficiently from the target tables.

**Apply program startup parameters important when tuning to maximize throughput**

Certain Apply program startup parameters affect replication performance. Use the following recommendations to maximize replication throughput.

For each of the startup parameters, there is a corresponding value in the IBMSNAP_APPPARMS table, although those values are not listed here. All startup parameters apply to both the asnapply and STRDPRAPY commands, unless otherwise stated.

**delay (for asnapply command)**

Set delay to 0 seconds or 1 second.

This parameter specifies the delay time (in seconds) at the end of each Apply cycle when continuous replication is used. Set to no delay for high-volume environments. For lower-volume environments that must still have low latency, set this value to 1 to minimize the number of connection requests from the Apply program to the source server.

**spillfile (for asnapply command)**

Set spillfile to mem.

This parameter specifies that the Apply program store its spill files in memory. If you do not specify sufficient memory for the Apply job by using the REGION keyword, the Apply program uses VIO. Storing the spill files in memory saves two I/O operations for every record that the Apply program processes.
Chapter 4. Tuning replication to minimize CPU usage

SQL replication must perform certain tasks to minimize CPU usage for replication.

To minimize CPU usage for replication, SQL replication must perform both of the following tasks:

- Capture changes from the DB2 log or journal and write these changes to the CD tables efficiently and only as often as necessary to maintain a minimum allowable replication latency.
- Fetch the changes from the CD tables and apply them to the target tables efficiently and only as often as necessary to maintain a minimum allowable replication latency.

When you tune replication for minimum CPU usage, you must often increase the overall replication latency. This section assumes that throughput and latency are not as important as CPU usage.

Tuning the Capture program to minimize CPU usage

There are certain general recommendations to follow when minimizing CPU usage for the Capture program. This section details general recommendations on how to minimize CPU usage for the Capture program.

Use the following general recommendations to minimize CPU usage for the Capture program:

- Consider capturing only changes made to registered columns.
  You can control whether the Capture program writes a row to a CD table for changed source rows according to the following row-capture rules:
  1. Capture changes only when registered columns change
  2. Capture changes when any column (registered or not) changes
  These rules correspond to the CHGONLY column of the IBMSNAP_REGISTER table.

If you register a subset of columns and usually nonregistered columns are updated in the source table, the Capture program can skip many rows when capturing data from the DB2 log, and thus decrease its CPU usage. For this case, specify the first row-capture rule (“Capture changes only when registered columns change”). However, if you register all columns, or if usually only registered columns are updated in the source table, do not specify the first row-capture rule because no rows (or few rows) can be skipped, and the Capture program’s CPU usage can increase unnecessarily. Check the statistics in the Capture monitor (IBMSNAP_CAPMON) table to verify whether the Capture program is able to skip rows because of the row-capture rule; see Chapter 5 “Measuring replication performance,” on page 23 for more information.

In the Replication Center, specify either of the row-capture rules: Capture changes to registered columns only or Capture changes to all columns. For the System i ADDDPFRREG command, specify either GENCDROW=*ALLCHG or GENCDROW=*REGCOLCHG.

- For z/OS environments, use dynamic SQL caching.
  Dynamic SQL caching allows the SQL PREPARE statement to be cached, and therefore reused when the identical statement is reissued.
For System i environments, register all columns of a source table and include only the afterimage columns in the CD table.

For example, if you have a source table that contains 100 columns, and you decide to save some disk space for the CD table by registering 94 of the source columns (perhaps because the other 6 columns are not needed for the target tables), you will discover that CPU usage for the Capture program will increase to filter the extra columns out for every insert into the CD table.

**Capture program startup parameters important to minimize CPU usage**

Certain Capture program startup parameters affect replication performance. Use recommendations detailed in this section to minimize CPU usage.

For each of the startup parameters, there is a corresponding value in the IBMSNAP_CAPPARMS table, although those values are not listed here. All startup parameters apply to both the asncap and STRDPRCAP commands, unless otherwise stated.

**commit_interval (for asncap command) or FRCFRQ (STRDPRCAP command)**

Set `commit_interval` or `FRCFRQ` to a high value between 30 and 60 seconds.

- **z/OS**
  - Linux
  - UNIX
  - Windows
  - This parameter specifies the approximate number of seconds that will elapse between Capture commit points. The smaller the value that you specify, the sooner changed data will be available to the Apply program, which fetches only data that has been committed. However, the smaller this value is, the greater the amount of overhead that is consumed by the Capture program. In addition to the DB2 commit itself, which causes a forced log write, there is the cost of re-preparing dynamic SQL statements. This cost increases as you register more tables for replication and thus use more CD tables. Higher values cause the Capture program to behave like a long-running batch job, which can impact DB2 restart should the Capture program fail.

  **Important:** Set this value to a value less than the DB2 lockout value to prevent contention between the Capture worker and pruning threads.

- **System i**
  - This parameter tells Capture how often to write rows to the CD tables, which is especially important for programs that do not use commit control.

**monitor_interval (for asncap command) or MONITV (for STRDPRCAP command)**

Set `monitor_interval` to a high value.

This parameter specifies how frequently (in seconds) the Capture program inserts rows into the Capture monitor (IBMSNAP_CAPMON) table. Setting a high value reduces the amount of overhead for gathering Monitor statistics.

- **System i**
  - An interval of less than 120 seconds is not supported in a System i environment. Slow performance could result if the interval is too small.

**prune_interval (for asncap command)**

Set `prune_interval` to a high value.
This parameter specifies how frequently (in seconds) the CD, unit-of-work (UOW), Capture monitor (IBMSNAP_CAPMON), Capture trace (IBMSNAP_CAPTRACE), and signal (IBMSNAP_SIGNAL) tables are pruned. Setting a high value reduces the overhead required for processing pruning queries during each pruning interval.

**sleep_interval (for asncap command)**
Set **sleep_interval** to a moderate value between 5 and 10 seconds.

- **Linux UNIX Windows** This parameter specifies the number of seconds that the Capture program sleeps when it finishes processing the active log and determines that the buffer is empty. Set this parameter to a moderate value to minimize CPU usage for lower volume environments.

- **z/OS** For z/OS data-sharing environments, this parameter specifies the number of seconds that the Capture program sleeps after the buffer returns less than half full.

---

**Tuning the Apply program to minimize CPU usage**

Because the Apply program is an SQL application, tune DB2 to minimize its CPU usage just as you tune any other SQL application.

For recommendations on tuning DB2 to minimize its CPU usage, review the following:

- “Linux, UNIX, and Windows database tuning” on page 3
- “Tuning Linux, UNIX, and Windows database parameter settings” on page 4
- “Tuning UNIX and Windows database environment-variable settings” on page 6
- “Database tuning for z/OS and USS” on page 6
- “Tuning z/OS and USS database parameter settings” on page 7
Chapter 5. Measuring replication performance

Use the Replication Center to monitor your replication environment and determine how the performance-tuning changes that you make affect your replication environment.

The Replication Center provides snapshot monitoring for the Capture and Apply programs, and it allows you to set alert conditions for the Replication Alert Monitor. You can:

- View statistics gathered by the Capture program.
  - Use the Capture Throughput Analysis window to view the number of rows that the Capture program inserted into the CD table or skipped (because of rows not recaptured in an update-anywhere environment, rows deleted by a CD-table trigger, or the row-capture rule “Capture changes only when registered columns change”) during each Monitor interval.
  - Use the Capture Throughput Analysis window to view the number of rows that the Capture program pruned from CD tables during each pruning interval.
  - Use the Capture Throughput Analysis window to view the number of transactions that the Capture program committed during each Monitor interval, including the transaction size and how many transactions spilled to disk because of memory overflow.
  - Use the Capture Throughput Analysis window to view how much memory the Capture program used during each Monitor interval.
  - Use the Capture Latency window to view the latency of the Capture program during each Monitor interval.

- Display performance information for the Apply program.
  - Use the Apply Throughput Analysis window to display the number of rows fetched from CD tables.
  - Use the Apply Throughput Analysis window to display the elapsed time for each subscription set.
  - Use the End-to-End Latency window to display a report on the average, minimum, maximum, and current latency for each subscription set.

- Define Capture alerts for the Replication Alert Monitor.
  - Define an alert for Status last committed: the maximum number of seconds between the times that Capture commits data to the CD table during a Monitor cycle.
  - Define an alert for Capture latency: the maximum number of seconds between the times that Capture commits data to the CD table.
  - Define an alert for Historic latency: the maximum number of seconds that the Capture latency can be out of the specified range.
  - Define an alert for Memory used: the amount of memory that the Capture program uses.

- Define Apply alerts for the Replication Alert Monitor.
  - Define an alert for Subscription sets delayed: the maximum number of seconds that a subscription set’s processing can be delayed (calculated by using columns from the IBMSNAP_SUBS_SET table: CURRENT_TIMESTAMP - LASTRUN > alert_value).
- Define an alert for End-to-end latency: the maximum number of seconds for end-to-end replication latency (calculated using columns from the IBMSNAP_APPLYTRAIL and IBMSNAP_SUBS_SET tables: (LAISTRUN - ENDTIME) + (SOURCE_CONN_TIME - SYNCHTIME)).
Product documentation

Documentation is provided in a variety of locations and formats, including in help that is opened directly from the product interface, in a suite-wide information center, and in PDF file books.

You can also order IBM publications in hardcopy format online or through your local IBM representative.

To order publications online, go to the IBM Publications Center at [www.ibm.com/shop/publications/order](http://www.ibm.com/shop/publications/order).

You can send your comments about documentation in the following ways:

- E-mail: comments@us.ibm.com

Contacting IBM

You can contact IBM for customer support, software services, product information, and general information. You can also provide feedback on products and documentation.

Customer support

For customer support for IBM products and for product download information, go to the support and downloads site at [www.ibm.com/support/us/](http://www.ibm.com/support/us/).

You can open a support request by going to the software support service request site at [www.ibm.com/software/support/probsub.html](http://www.ibm.com/software/support/probsub.html).

My IBM

You can manage links to IBM Web sites and information that meet your specific technical support needs by creating an account on the My IBM site at [www.ibm.com/account/us/](http://www.ibm.com/account/us/).

Software services


Information Management product support

For Information Management product support, news, and other product information, go to the Information Management support site at [www.ibm.com/software/data/support/](http://www.ibm.com/software/data/support/).

Federation, replication, and event publishing products support

For support, go to:

• IBM InfoSphere Replication Server
  www.ibm.com/software/data/integration/support/replication_server/
• IBM InfoSphere Data Event Publisher
  www.ibm.com/software/data/integration/support/data_event_publisher/

Classic products support

For support, go to:
• IBM InfoSphere Classic Federation Server for z/OS
  www.ibm.com/software/data/integration/support/classic_federation_server_z/
• IBM InfoSphere Classic Replication Server for z/OS
  www.ibm.com/software/data/infosphere/support/replication-server-z/
• IBM InfoSphere Classic Data Event Publisher for z/OS
  www.ibm.com/software/data/integration/support/data_event_publisher_z/
• IBM InfoSphere Data Integration Classic Connector for z/OS
  www.ibm.com/software/data/integration/support/data_integration_classic_connector_z/

General information

To find general information about IBM, go to www.ibm.com.

Product feedback

You can provide general product feedback through the Consumability Survey at

Documentation feedback

You can click the feedback link in any topic in the information center to comment
on the information center.

You can also send your comments about PDF file books, the information center, or
any other documentation in the following ways:
• Online reader comment form: www.ibm.com/software/data/rcf/
• E-mail: comments@us.ibm.com
How to read syntax diagrams

The following rules apply to the syntax diagrams that are used in this information:

- Read the syntax diagrams from left to right, from top to bottom, following the path of the line. The following conventions are used:
  - The >> symbol indicates the beginning of a syntax diagram.
  - The ---> symbol indicates that the syntax diagram is continued on the next line.
  - The >-- symbol indicates that a syntax diagram is continued from the previous line.
  - The --->< symbol indicates the end of a syntax diagram.
- Required items appear on the horizontal line (the main path).

  >>>---required_item

- Optional items appear below the main path.

  >>>---required_item---optional_item

If an optional item appears above the main path, that item has no effect on the execution of the syntax element and is used only for readability.

  >>>---required_item---optional_item

- If you can choose from two or more items, they appear vertically, in a stack.
  If you must choose one of the items, one item of the stack appears on the main path.

  >>>---required_item---required_choice
  >>>---required_choice

If choosing one of the items is optional, the entire stack appears below the main path.

  >>>---required_item---optional_choice
  >>>---optional_choice

If one of the items is the default, it appears above the main path, and the remaining choices are shown below.

  >>>---required_item---default_choice
  >>>---optional_choice

- An arrow returning to the left, above the main line, indicates an item that can be repeated.
If the repeat arrow contains a comma, you must separate repeated items with a comma.

A repeat arrow above a stack indicates that you can repeat the items in the stack.

- Sometimes a diagram must be split into fragments. The syntax fragment is shown separately from the main syntax diagram, but the contents of the fragment should be read as if they are on the main path of the diagram.

```
required_item  fragment-name
```

**Fragment-name:**

- Keywords, and their minimum abbreviations if applicable, appear in uppercase. They must be spelled exactly as shown.
- Variables appear in all lowercase italic letters (for example, column-name). They represent user-supplied names or values.
- Separate keywords and parameters by at least one space if no intervening punctuation is shown in the diagram.
- Enter punctuation marks, parentheses, arithmetic operators, and other symbols, exactly as shown in the diagram.
- Footnotes are shown by a number in parentheses, for example (1).
Product accessibility

You can get information about the accessibility status of IBM products.

The IBM InfoSphere™ Information Server product modules and user interfaces are not fully accessible. The installation program installs the following product modules and components:
- IBM InfoSphere Business Glossary
- IBM InfoSphere Business Glossary Anywhere
- InfoSphere DataStage® and QualityStage Administrator
- IBM InfoSphere FastTrack
- IBM InfoSphere Information Analyzer
- IBM InfoSphere Information Services Director
- IBM InfoSphere Metadata Workbench

For more information about IBM product accessibility status, go to [http://www.ibm.com/able/product_accessibility/index.html](http://www.ibm.com/able/product_accessibility/index.html)

Accessible documentation

Accessible documentation for IBM Information Server products is provided in an information center. The information center presents the documentation in XHTML 1.0 format, which is viewable in most Web browsers. XHTML allows you to set display preferences in your browser. It also allows you to use screen readers and other assistive technologies to access the documentation.
Notices

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 character set (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.
3-2-12, Roppongi, Minato-ku, Tokyo 106-8711 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 Corporation
J46A/G4
555 Bailey Avenue
San Jose, CA 95141-1003 U.S.A.

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

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

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

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

**Trademarks**

IBM trademarks and certain non-IBM trademarks are marked on their first occurrence in this information with the appropriate symbol.

IBM, the IBM logo, and ibm.com 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](http://www.ibm.com/legal/copytrade.shtml).

The following terms are trademarks or registered trademarks of other companies:

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency, which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

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

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

ITIL is a registered trademark and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

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

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.
Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.
Index

A
accessibility 25
ADDDPRREG command 19
aggregate table 9
answer set
large 16
Apply fetch rate 13
Apply program 1
  Apply program
    alerts 23
    maximize throughput 18
  Capture program
    alerts 23
    maximize throughput 17, 18
    minimize CPU usage 21
    multiple instances 17
    parameters 18
    performance information 23
    performance tuning 18
    pull configuration 10
    remotely connected 11
    Replication Alert Monitor 23
    startup parameters 11
    tuning
      maximize throughput 16, 17
Apply Throughput Analysis window 23
apply_path 3
asapply command 18
asncap command 11, 14, 20
asncpl command 9, 18
ASNCLP program 1
asntroff command 11
autopruner parameter 11

B
bandwidth, increase 17
BLKSIZE database parameter setting 7
BLKSIZE parameter 7
bufferpool 13
bufferpool overhead 11
bufferpools 3, 6

C
CACHE parameter setting 7
CACHE parameter 7
Capture changes 13
Capture Latency Capture alert 23
Capture Latency window 23
Capture program
  maximize throughput 14
  minimize CPU usage 19
  parameters 14
    for CPU usage 20
    performance tuning 20
    startup parameters 11
    statistics 23
Capture program (continued)
  tuning
    maximize throughput 14
to maximize throughput 13, 14
Capture program throughput 11
Capture schemas
  maximize throughput 14
  tuning 14
Capture Throughput Analysis
  window 23
CD tables 9
CLNUPITV parameter 11
commit_interval parameter 14, 20
conflict detection 9
CPU usage 9
  minimize 19, 20
  create subscription sets 8
  creating subscription sets 18
  customer support 25
D
data blocking 16
data-sharing considerations
  USS 11
  z/OS 11
database environment variables 3, 6
database environment-variable settings
  UNIX 6
  Windows 6
database objects 6
  Linux, UNIX, and Windows 3
  USS 6
  z/OS 6
database parameters 3, 6
database tuning
  Linux, UNIX, and Windows 3
  USS 6
  z/OS 6
DataPropagator 10
DB2_MINIMIZE_LIST_PREFETCH
  variable 6
DB2_MMAP_READ variable 6
DB2_PARALLEL_IO variable 6
DBHEAP parameter 4
DEALLCT database parameter setting 7
DEALLCT parameter 7
delay parameter 18
delete data more efficiently 18
disk systems 6
documentation
  accessible 25
DRDA protocol 10
DSNZPRM replication settings 7
DSNZPRM values 7
dynamic SQL caching 19

E
End-to-end latency
  Apply alert 23
End-to-End Latency window 23
ENDDPRCAP command 9
environment variables for replication 6
environment-variable settings 6
evaluating performance-tuning changes 23

F
fetch changes 13
FRCFRQ parameter 14, 20

H
Historic latency Capture alert 23
IBM support 25
IBMSNAP_APPLYTRAIL table 23
IBMSNAP_APPPARMS table 11, 18
IBMSNAP_CAPPARMS table 11
IBMSNAP_REGISTER table 19
IBMSNAP_SUBS_MEMBR table 9
IBMSNAP_SUBS_SET table 23
increase resource consumption 13
index
  optimized 18
  indexes 18
  fewer 18
insert data more efficiently 18
INTRA_PARALLEL parameter 4
Introduction
  SQL replication 1

J
job
  priority 17

L
latency 9, 13
legal notices 31
Linux, UNIX, and Windows
  parameter settings 4
LOB columns 17
LOCKLIST parameter 4
LOGBUFFS parameter 4
LOGFILSZ parameter 4
logical unit (LU) name 11
LOGPRIMARY parameter 4
logs 3, 6
LOGSECOND parameter 4
LU names 11
maximize replication throughput 18
throughput 8, 11, 13, 14, 16
maximize throughput 17, 18
MAXLOCKS parameter 4
MAXRTU database parameter setting 7
MAXRTU parameter 7
measuring performance 23
MEMLMT parameter 11
Memory used Capture alert 23
memory_limit parameter 11
minimize CPU usage 11, 19, 20, 21
processor usage 8
minimize replication latency 19
monitor performance-tuning 23
monitor_interval parameter 14, 20
MONITV parameter 14, 20
multiple Capture schemas 13
nodegroup 14
NUM_IOCLEANERS parameter 4
NUM_IOSERVERS parameter 4
opt4one parameter 11
optimize CD tables 9
UOW table 9
OPTSNGSET parameter 11
OS/400 20
OUTBUFF database parameter setting 7
OUTBUFF parameter 7
parameter settings Linux, UNIX, and Windows 4
USS 7
z/OS 7
PCKCACHESZ parameter 4
peer-to-peer replication 10
performance tuning 1, 3, 4, 6, 8, 9, 10, 11, 13, 14, 16, 17, 18, 19, 21, 23
performance tuning with replication trace 11
point-in-time table 9
predicates 9
PREFORMAT option 9
priority Capture program 14
job 14, 17
process 14
processor usage minimize 8
product accessibility accessibility 29
prune_interval parameter 20
pruning 1
pull configuration 10
read-only target table 10
recommended parameters 7
reduce I/O 13
register all columns 19
register source tables 8
registered columns 19
remote journal benefits 10
remote journals 17
REORG utility 9
replication continuous 16
update-anywhere 9
Replication Alert Monitor 1
Replication Center 1
replication latency minimize 19
replication trace 11
replication tuning 7
replication tuning goals 8
RGZCCTLTBL keyword 9
RGZPFM command 9
row-capture rules 19
Run Priority field 14, 17
RUNSTATS utility 3, 6, 9
screen readers 25
setpri command 14, 17
sleep_interval parameter 14, 20
software services 25
spill file 10
spill files 3, 6, 17
spillfile parameter 18
SQL replication major components 1
SQL script 9
SQL scripts 18
startup parameters Apply program 11
start-up parameters Capture program 11
statistics 3, 6
Status last committed Capture alert 23
STRDYPRAPY command 11, 18
STRDYPRCAP command 11, 14, 20
subscription set definition 16
group multiple members 16
subscription sets 9
Subscription sets delayed Apply alert 23
subscription-set member 9
support, customer 25
System i 1, 9, 10, 11, 14, 16, 17, 18, 19
System i system commands 1
table spaces 3
throughput maximize 8, 11, 13, 14, 16
throughput maximization 17, 18
throughput, maximization 17
trace 11
trademarks 33
tuning databases Linux, UNIX, and Windows 3
USS 6
z/OS 6
tuning preliminaries 6
turn replication trace off 11
UNIT database parameter setting 7
UNIT parameter 7
UNIX database environment-variable settings 6
UOW table 9
UOW_CD_PREDICATES column 9
update-anywhere replication 9
update-anywhere environment 17
update-anywhere replication 10
user-copy table 9
USS data-sharing considerations 11
parameter settings 7
z/OS data-sharing considerations 11
parameter settings 7