

Understanding IBM Db2 Restore

Creating regular backups is a critical step to recovering from a disaster. Restoring the database is arguably the most important and time-critical portion of the backup and restore process. This IBM® Redbooks® web doc describes the following elements that are critical to a successful restore process:

- The process model
- Restore to existing and non-existing databases
- Restore individual table spaces
- Restore a subset of table spaces to a new database
- Change the database path
- Changing the container path
- Considerations for the Database Partitioning Feature (DPF)
- Monitoring the restore operation progress

This document applies to all versions of IBM Db2® Linux, UNIX, and Windows.

The process model

The restore process model consists of the engine dispatchable units (EDU):

- db2agent
- db2bm
- db2med
- db2vend (if the backup image originates from a storage manager)

The db2agent EDU does most of its work during the start of the restore process, and then it waits for the restore process to complete. These are the tasks it performs:

- Calculates and spawn an optimal number of db2bm EDUs
- Spawns the db2med EDU
- Allocates an optimal number of buffers, compatible with the buffer size used during the backup
- Displays the selected tunable values to the db2diag.log
- Allocates message queues for inter-EDU communication
- Re-creates the database configuration file if the target database does not exist
- Coordinates the backup process by sending control messages to the db2bm and db2med EDUs

The db2med EDU acquires a buffer from the empty queue and fills the buffer with a read from the backup image. The db2med EDU then places the buffer on the full queue to be written to disk by a db2bm EDU. Only one db2med EDU is created even if multiple parts exist in the backup image.

The db2bm EDU acquires a buffer from the full queue. The buffer header indicates which table space this buffer belongs to and the page offset in the table space where this buffer belongs. Next, the db2bm writes the contents of the buffer to the appropriate container. If multiple containers are in the table space, the I/O is divided into extents and written to disk in parallel by the prefetcher EDU. The db2bm EDU places the buffer in the empty queue for reuse by the db2med EDU.

The db2vend process is created if the backup image is being retrieved from a storage manager. The db2vend process uses a vendor API to read data from the backup image and forwards the data to the db2med EDU.

Figure 1 graphically represents the restore process model.

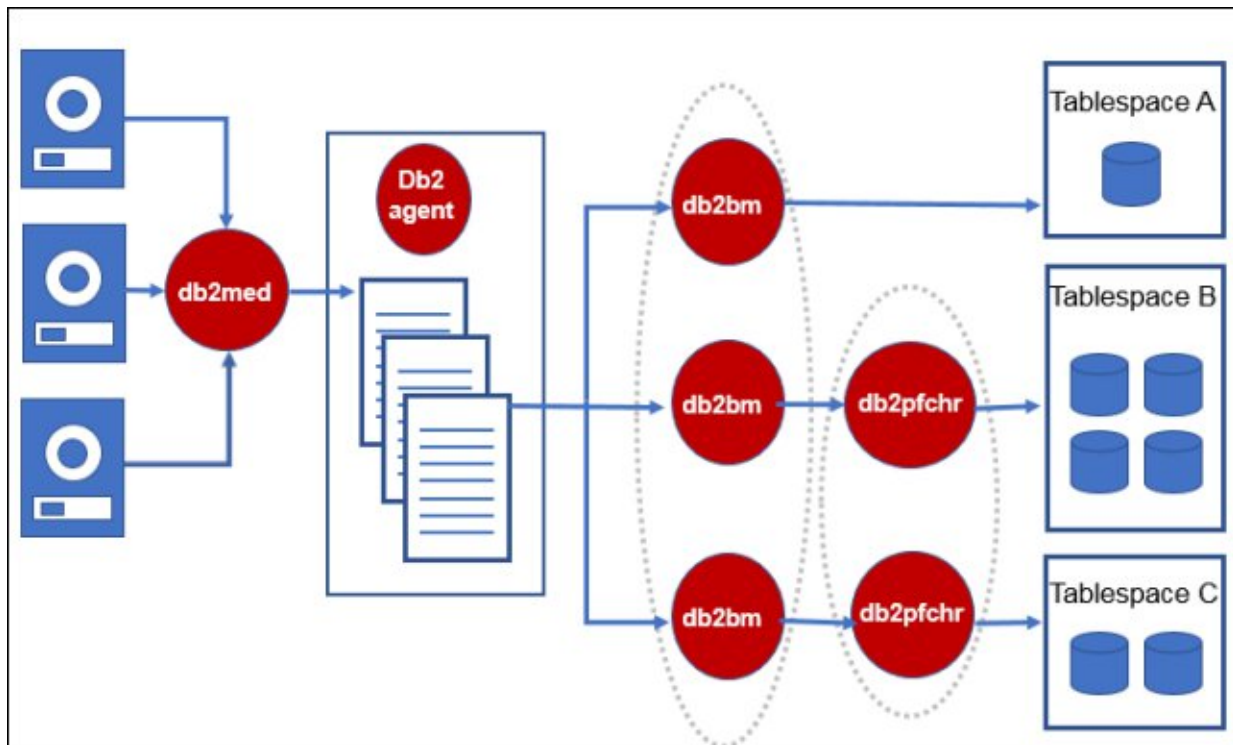


Figure 1. Restore process model

Restore to existing and non-existing databases

Key differences exist between restoring to an existing database and restoring to a non-existing database. One of the most critical details is which database configuration settings are used. Specifically the database configuration settings for logging and self-tuning memory might be different and cause unexpected results.

When restoring to an existing database, consider these factors:

- The database configuration file from the existing database is used.
- Containers from the existing database are reused and resized as necessary.
- Extra containers on an existing database but not in backup image are deleted.
- The restore operation prompts for confirmation when replacing a database by the same name.

For a complete list of other actions performed during a restore operation, see "Restoring to an existing database" (Db2 11.1 for Linux, UNIX, and Windows) in IBM Knowledge Center:

<https://ibm.biz/BdjB83>

When restoring to a non-existing database, consider these factors:

- The database configuration file from the database in the backup image is used.
- Containers are created serially by the db2bm EDU.

The time required to create the containers varies depending on the size of the containers and if fast preallocation is available. Fast preallocation availability depends on the operating system (OS) and file system type. To determine if fast preallocation is available, review the setting of the DB2_USE_FAST_PREALLOCATION variable. For details, see "Performance Variables" (Db2 11.1 for Linux, UNIX, and Windows) in IBM Knowledge Center: <https://ibm.biz/BdjB8k>

Restore individual table spaces

To restore an individual table space, several conditions must be met (see the commands and responses in Example 1):

- Archival logging must be enabled.
- Restored table spaces are in a rollforward pending status.
- Rollforward to end of log is required to remove the roll forward pending status.
- The entire contents of all backup images is read.

Example 1. Commands and responses to restore an individual table space

```
db2 restore database sample tablespace (ts1)
db2pd -db sample -tab 3
```

Tablespace 3 Configuration:

| Address | Type | Content | PageSz | <...> | Name |
|--------------------|------|---------|--------|-------|------|
| 0x00007FFEB871D440 | DMS | Large | 4096 | <...> | TS1 |

Tablespace 3 Statistics:

| Address | TotalPgs | UsablePgs | UsedPgs | <...> | State |
|--------------------|----------|-----------|---------|-------|------------|
| 0x00007FFEB871D440 | 16384 | 16352 | 11648 | <...> | 0x00000080 |

```
db2tbst 0x80
State = Roll Forward Pending
```

```
db2 rollforward db sample to end of logs and complete tablespace (ts1) online
db2pd -db sample -tab 3
```

Tablespace 3 Configuration:

| Address | Type | Content | PageSz | <...> | Name |
|--------------------|------|---------|--------|-------|------|
| 0x00007FFEB871D440 | DMS | Large | 4096 | <...> | TS1 |

Tablespace 3 Statistics:

| Address | TotalPgs | UsablePgs | UsedPgs | <...> | State | MinRecTime | NQuiescers | PathsDropped |
|--------------------|----------|-----------|---------|-------|-------------|------------|------------|--------------|
| 0x00007FFEB871D440 | 16384 | 116352 | 11648 | <...> | 00x00000000 | 1 | | |

Restore a subset of table spaces to a new database

Restoring a subset of table spaces in the backup image is possible by using the REBUILD clause of the restore database command. The following options are available for the REBUILD clause. In all cases, a restore database command with the REBUILD clause overwrites the existing database. Table spaces excluded from the restore process are still part of the database and are in a restore pending status.

- REBUILD WITH ALL TABLESPACES IN DATABASE
Restores the database with all the table spaces known to the database at the time of the image being restored.
- REBUILD WITH ALL TABLESPACES IN DATABASE EXCEPT <table space list>
Restores the database with all the table spaces known to the database at the time of the image being restored except for those specified in the list.

- **REBUILD WITH ALL TABLESPACES IN IMAGE**
Restores the database with only the table spaces in the image being restored.
- **REBUILD WITH ALL TABLESPACES IN IMAGE EXCEPT <table space list>**
Restores the database with only the table spaces in the image being restored except for those specified in the list.
- **REBUILD WITH <table space list>**
Restores the database with only the list of table spaces specified.

Sample **db2 restore** commands with the **REBUILD** clause are as follows:

- `db2 restore db sample rebuild with all tablespaces in database`
- `db2 restore db sample rebuild with all tablespaces in database except tablespace (ts1,ts2)`
- `db2 restore db sample rebuild with all tablespaces in image`
- `db2 restore db sample rebuild with all tablespaces in image except tablespace (ts1,ts2)`
- `db2 restore db sample rebuild with tablespace (syscatspace,ts1,ts2)`

Change the database path

The database path indicates where the database control files are located and the default location for the active log files. This path can be changed only if the database does not exist prior to the restore command.

To change the database path, these options of the **restore database** command are available:

- **TO**
This option is the default. If the database is enabled for automatic storage, only the database path is changed, and the container paths are not changed. Here is an example:
`db2 restore db sample to /dbpath`
- **DBPATH ON**
This option is used in conjunction with the **ON** clause. If the database is enabled for automatic storage, then the database path is changed with the **DBPATH ON** clause and the container paths are changed with the **ON** clause. Here is an example:
`db2 restore db sample on /data1 dbpath on /dbpath`

Change the container path

How you change container path names depends on whether or not you are using automatic storage:

- If you use automatic storage: Changing container path names is done with the **ON** clause of the restore statement or with a redirected restore.
- If you do not use automatic storage: Changing container paths is done with a redirected restore.

For details, see "Performing a redirected restore operation" (Db2 11.1 for Linux, UNIX, and Windows) in IBM Knowledge Center:

<https://ibm.biz/BdjB8t>

The following command changes the automatic storage definition for the database and creates the restored containers on the three specified paths:

```
db2 restore db sample on /data1, /data2, /data3
```

You can also use a redirected restore option as shown in the following examples:

- `db2 restore database sample redirect`
- `db2 set stogroup paths for ibmstogroup on '/data1', '/data2'`
- `db2 restore database sample continue`

A non-automatic storage redirected restore is shown in Example 2. The `set tablespace containers` command must be run for every table space that requires a new container path.

Example 2. Non-automatic storage redirected restore

```
db2 restore database sample redirect
db2 set tablespace containers for 0 using ( path '/data1/sqlt0000.0' )
db2 set tablespace containers for 1 using ( path '/data1/sqlt0001.0' )
db2 set tablespace containers for 2 using ( path '/data1/sqlt0002.0' )
db2 set tablespace containers for 3 using ( file '/data1/ts1' 10000 )
db2 restore database sample continue
```

Considerations for the Database Partitioning Feature (DPF)

When using the DPF feature of Db2, the restore process can become complex. To provide transactional consistency between members, all members must be rolled forward to the same point in time, as follows:

- When circular logging is used, a restore must be performed on all members by using the offline backup for each member with the same backup timestamp.
- When using archival logging, restoring an individual member is possible if the transaction logs can be applied in order to bring the restored member up-to-date with all other members, with a roll forward to the end of logs.
- The `restore database` command must be executed individually on each member.
- The `rollforward database` command must be executed from the catalog member.

Examples 3 - 6 show restore and rollforward operations. The sample database was created across three members and the backup was taken by using the command shown in the examples.

Example 3. Backup database command

```
db2 backup db sample on all dbpartitionnums
```

```
Part  Result
```

```
-----
0000  DB20000I  The BACKUP DATABASE command completed successfully.
0001  DB20000I  The BACKUP DATABASE command completed successfully.
0002  DB20000I  The BACKUP DATABASE command completed successfully.
```

Backup successful. The timestamp for this backup image is : 20170614135931

Example 4. Restore database on all members and a rollforward to end of backup

```
db2_all "|db2 restore database sample from /db2backup taken at 20170614135931
without prompting"
db2 "rollforward db sample to end of backup and complete"
```

Example 5. Restore database on member 1 and a rollforward to end of logs

```
db2_all "<<+1< db2 restore database sample from /db2backup taken at
20170614135931 without prompting"
db2 "rollforward db sample to end of logs on dbpartitionnum (1) and complete"
```

Example 6. Restore table space ts1 online on member 1 and a rollforward to the end of the logs

```
db2_all "<<+1< db2 restore database sample tablespace \ (ts1\ ) online from
/db2backup taken at 20170614135931 without prompting"
db2 "rollforward db sample to end of logs on dbpartitionnum (1) and complete
tablespace (ts1) online"
```

Monitoring the restore operation progress

To monitor the progress of the restore operation, you can use `db2pd -utilities`. While the restore operation is running, output from `db2pd -utilities` produces results (Example 7).

Example 7. Results of using `db2pd -utilities`

```
Utilities:
Address          ID  Type   State Invoker Priority StartTime          DBName NumPhases
CurPhase Description
0x000000020382DE00 190 RESTORE 0    0      0      Tue Jun 13 18:56:50 SAMPLE 1      1
db

Progress:
Address          ID  PhaseNum CompletedWork   TotalWork StartTime          Description
0x000000020382E128 190 1      2634670080 bytes Unknown   Tue Jun 13 18:56:50 n/a
```

By collecting multiple iterations of this output at known time intervals, the `TotalWork` values can be compared to determine the number of bytes per second being processed. Example 8 shows use of the `-repeat 10 2` option, which causes the command to be repeated twice, 10 seconds apart.

Example 8. Using the `-repeat 10 2` option of `db2pd`

```
db2pd -utilities -repeat 10 2
```

```
Database Member 0 -- Active -- Up 8 days 02:27:33 -- Date
2017-06-13-18.57.05.035735
```

```
Progress:
Address          ID  <...> CompletedWork   TotalWork <...>
0x000000020382E128 190      1728479232 bytes Unknown   <...>
```

```
Database Member 0 -- Active -- Up 8 days 02:27:43 -- Date
2017-06-13-18.57.15.048933
```

```
Progress:
Address          ID  <...> CompletedWork   TotalWork <...>
0x000000020382E128 190      2634670080 bytes Unknown   <...>
```

2634670080 - 1728479232 = 906190848 bytes in 10 seconds or 90619084.8 bytes per second.

By knowing how big the backup image is, the time remaining for the restore can be computed. The `TotalWork` column does not have the size of the complete backup image so it must be obtained by looking at the `db2diag.log` file from when the backup was created (Example 9).

Example 9. The db2diag.log file from backup

FUNCTION: DB2 UDB, database utilities, sqluxLogDataStats, probe:396
MESSAGE : Performance statistics
DATA #1 : String, 1414 bytes

Parallelism = 10
Number of buffers = 10
Buffer size = 16781312 (4097 4kB pages)

| BM# | Total | I/O | MsgQ | WaitQ | Buffers | kBytes |
|-----|--------|--------|------|-------|---------|---------|
| 000 | 37.80 | 23.03 | 0.84 | 13.67 | 33 | 507616 |
| 001 | 37.76 | 33.15 | 0.23 | 4.21 | 31 | 506624 |
| 002 | 37.76 | 30.30 | 0.47 | 6.85 | 31 | 506624 |
| 003 | 37.76 | 23.30 | 0.88 | 12.96 | 31 | 506624 |
| 004 | 37.76 | 29.10 | 0.45 | 7.54 | 31 | 506624 |
| 005 | 37.76 | 37.41 | 0.13 | 0.07 | 31 | 506624 |
| 006 | 37.77 | 33.23 | 0.25 | 4.14 | 31 | 506624 |
| 007 | 37.78 | 28.79 | 0.60 | 8.24 | 31 | 506624 |
| 008 | 37.77 | 33.92 | 0.39 | 3.30 | 31 | 506624 |
| 009 | 37.77 | 23.69 | 1.21 | 12.79 | 13 | 206976 |
| TOT | 377.75 | 295.96 | 5.50 | 73.82 | 294 | 4767584 |

| MC# | Total | I/O | MsgQ | WaitQ | Buffers | kBytes |
|-----|-------|------|-------|-------|---------|---------|
| 000 | 63.91 | 7.83 | 28.68 | 0.00 | 295 | 4801716 |
| TOT | 63.91 | 7.83 | 28.68 | 0.00 | 295 | 4801716 |

The total backup size is displayed in kilobytes (KB). From this backup image, the backup was 4801716 KB. To compute the time remaining for the restore operation (Example 10), subtract the CompletedWork value (seen in Example 8) from the size of the backup. This action requires converting the size of the backup into bytes so both measurements use the same units.

Example 10. Calculate time remaining to restore

$(4801716 * 1024) - 2634670080 = 2634670080$ bytes remaining in the restore.
 $2634670080 \text{ bytes} / 90619084.8 \text{ bytes per second} = 25.19$ seconds remaining in the restore.

Related information

IBM Knowledge Center has more information about the following commands:

- RESTORE DATABASE command:
<https://ibm.biz/BdjdXC>
- ROLLFORWARD DATABASE command:
<https://ibm.biz/BdjdXQ>

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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, MD-NC119, Armonk, NY 10504-1785, US

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 jurisdictions 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 websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

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

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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.

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 actual people or business enterprises 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.

© Copyright International Business Machines Corporation 2017. All rights reserved.

This document was created or updated on August 9, 2017.

Send us your comments in one of the following ways:

- Use the online **Contact us** review form found at:
ibm.com/redbooks
- Send your comments in an e-mail to:
redbooks@us.ibm.com
- Mail your comments to:
IBM Corporation, International Technical Support Organization
Dept. HYTD Mail Station P099
2455 South Road
Poughkeepsie, NY 12601-5400 U.S.A.

This document is available online at <http://www.ibm.com/redbooks/abstracts/tips1353.html> .

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

| | | |
|------|-----------|------------------|
| Db2® | IBM® | Redbooks (logo)® |
| DB2® | Redbooks® | |

The following terms are trademarks of other companies:

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.
Windows, and the Windows logo 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 company, product, or service names may be trademarks or service marks of others.