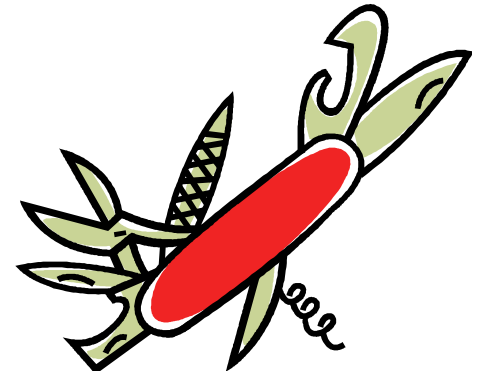


ADMIN_MOVE_TABLE

The Swiss Army Knife for Database Administrators



Beck Tang, IBM Canada
Zahidul Khan, IBM Canada
Phil King, IBM Canada

Special guests:
Jens Seifert, IBM Germany (Inventor of AMT)
Martin Jungfer, IBM Germany (Current maintainer of AMT)



Please Note

- IBM's statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM's sole discretion.
- Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.
- The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.
- The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.
- Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Acknowledgements and Disclaimers

Availability. References in this presentation to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates.

The workshops, sessions and materials have been prepared by IBM or the session speakers and reflect their own views. They are provided for informational purposes only, and are neither intended to, nor shall have the effect of being, legal or other guidance or advice to any participant. While efforts were made to verify the completeness and accuracy of the information contained in this presentation, it is provided AS-IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, this presentation or any other materials. Nothing contained in this presentation is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software.

All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual environmental costs and performance characteristics may vary by customer. Nothing contained in these materials is intended to, nor shall have the effect of, stating or implying that any activities undertaken by you will result in any specific sales, revenue growth or other results.

© Copyright IBM Corporation 2026. All rights reserved.

— **U.S. Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.**

IBM, the IBM logo, [ibm.com](https://www.ibm.com), DB2 and AIX are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or TM), these symbols indicate U.S. 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

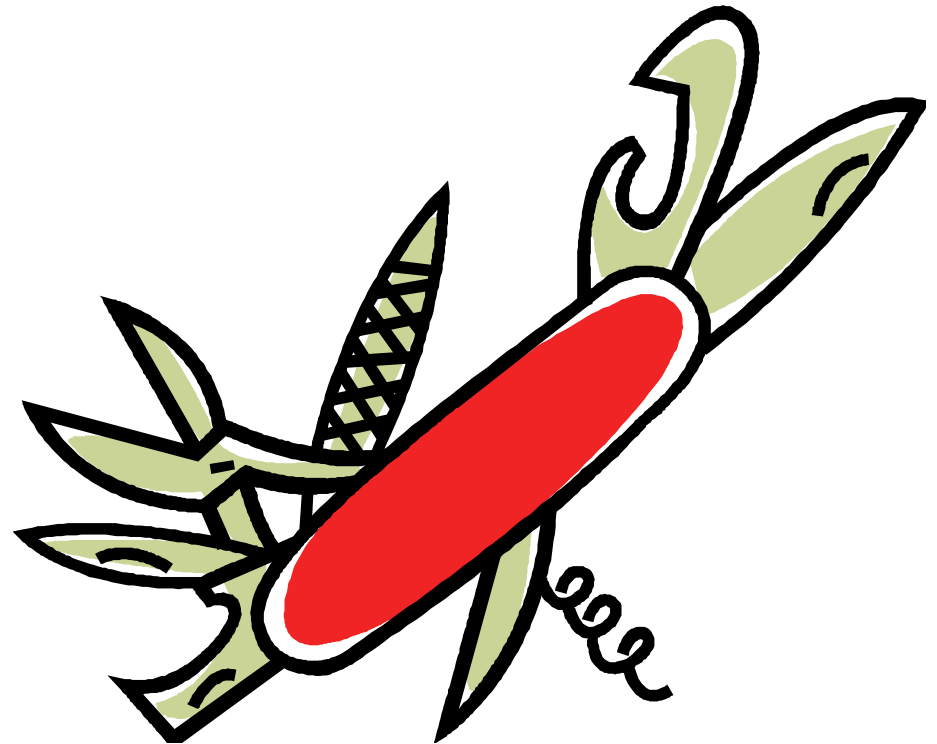
•“Copyright and trademark information” at <https://www.ibm.com/legal/copytrade>

•SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. Please see <https://www.sap.com/about/legal/copyright.html> and <https://www.sap.com/corporate/en/legal/trademark.html> for additional trademark information and notices.

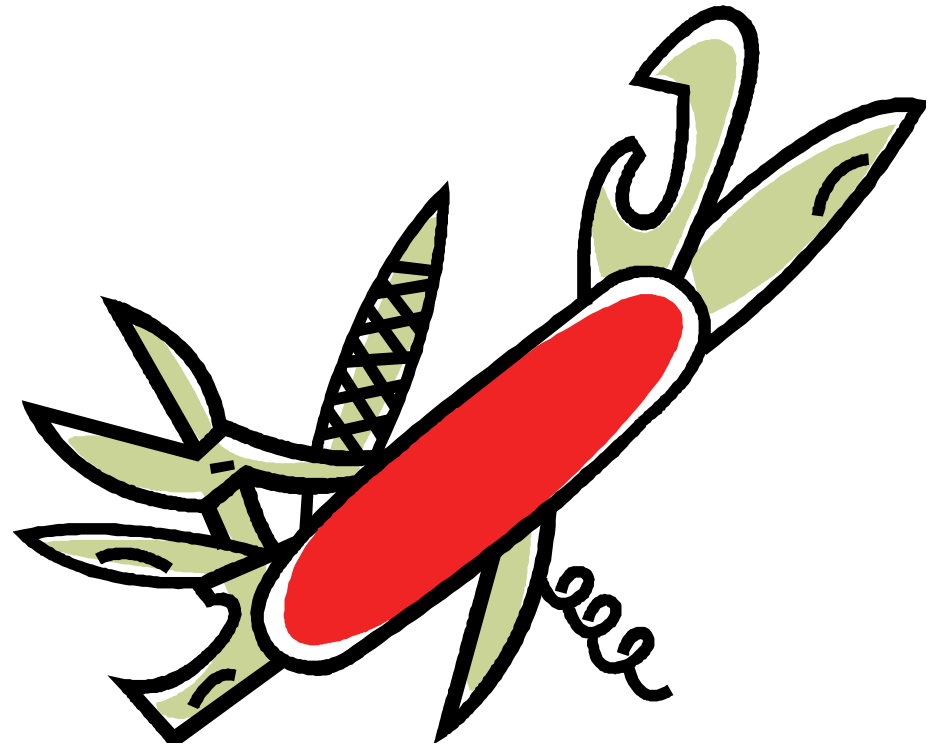
•Other company, product, or service names may be trademarks or service marks of others.

Agenda

- Introduction
- Use cases
- Features
- Performance
- Known limitations
- References



Introduction



The DBA's Challenge

Managing Table Operations

"We need to move this 500GB table to faster storage..."

But we can't have downtime

Business requires continuous availability

Maintenance windows are rare or non-existent

"Our compression isn't working well anymore..."

Table is 40% compressed, should be 70%

Wasting hundreds of GB

Can't take the app offline to fix it

"We need columnar storage for analytics..."

Queries too slow on row-organized tables

Users require constant access

Can't afford extended downtime for conversion

Traditional Approaches

The Availability Challenge

Move table to different storage

Traditional: Export → Import

X Hours of offline time for the tables

Fix poor compression

Traditional: REORG RESETDICTIONARY

X Table locked during operation

Convert to columnar (BLU)

Traditional: Manual export/import

X Complex, risky, offline

Change table structure

Traditional: DROP → CREATE → LOAD

X Application outage required

The Solution: ADMIN_MOVE_TABLE

What if you could do all of this... with minimal disruption?

- ✓ Move tables between tablespaces
- ✓ Fix compression issues
- ✓ Convert to columnar storage
- ✓ Change table structures
- ✓ Redistribute data

With brief table lock during SWAP (typically seconds)

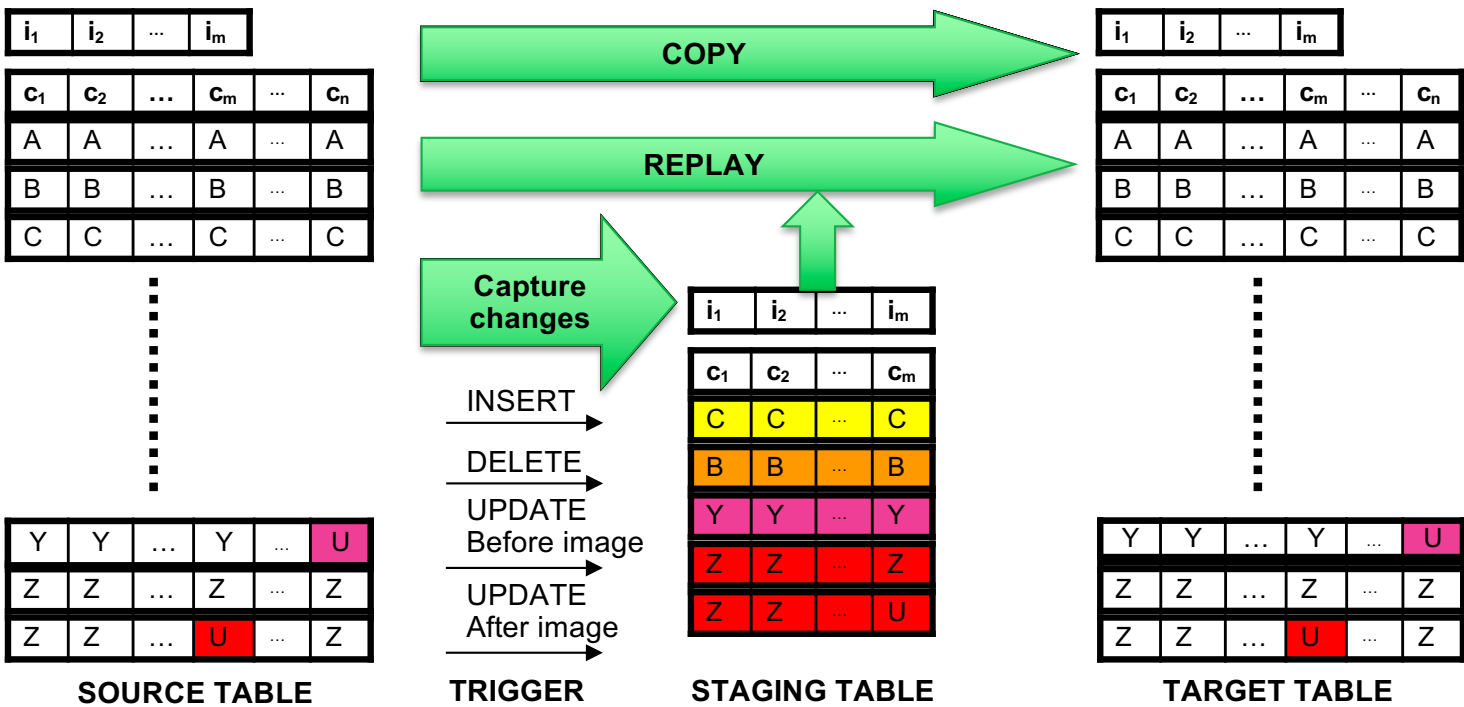
- Users can continue working (SELECT, INSERT, UPDATE, DELETE)
- Seconds of exclusive lock during final switch (SWAP phase)
- Fully recoverable – unless using nonrecoverable load

Available since Db2 9.7 (2009) • Enhanced in 11.5 and 12.1 • Known as ONLINE_TABLE_MOVE for SAP customers

General Concept – row-organized

Protocol table SYSTOOLS.ADMIN_MOVE_TABLE

tabschema	tabname	key	value
SAPR3	UTAB	INDEX_SCHEMA	SAPR3
...			



Selection of replay index

- Replay index is used in REPLAY of row-organized tables
- Replay index determines the columns of the staging table

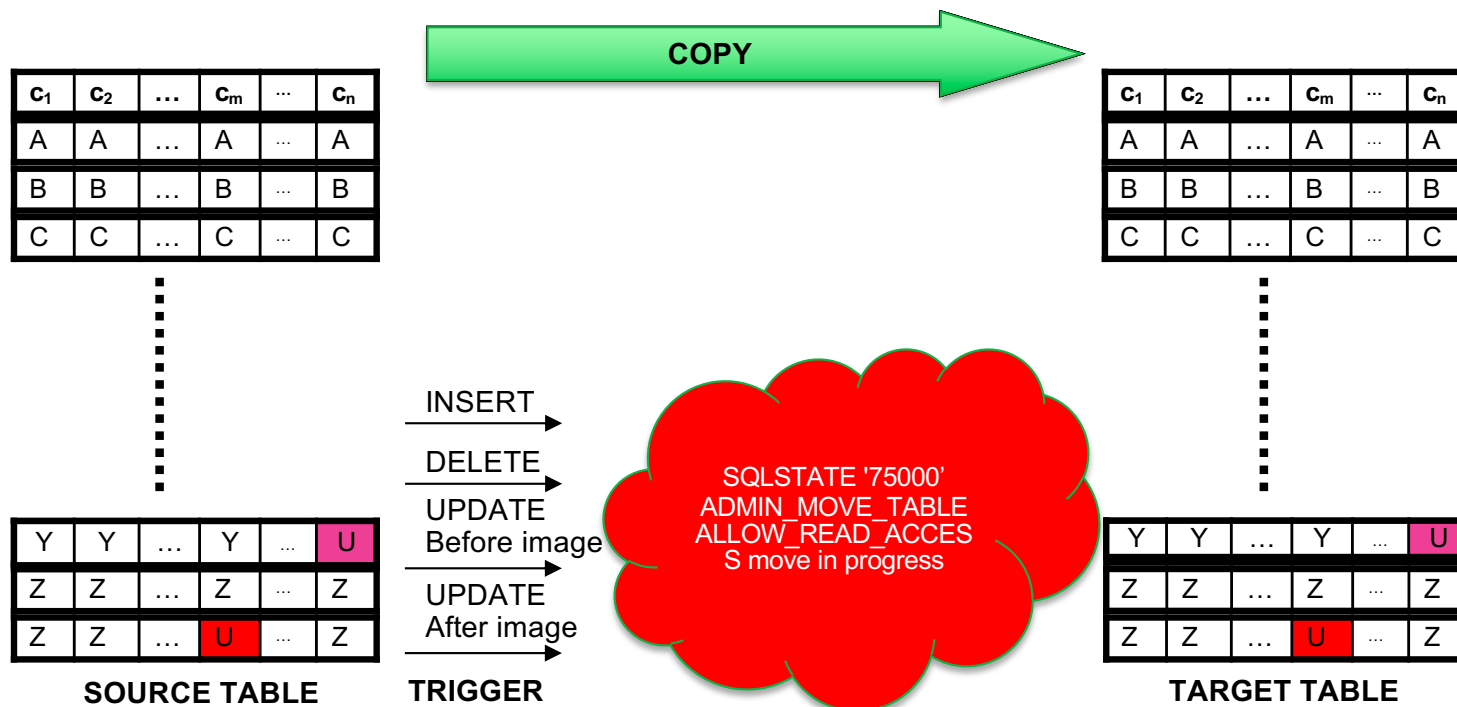
Index selection is done automatically by ADMIN_MOVE_TABLE:

1. PRIMARY index
 2. UNIQUE index
 3. Non-unique index with best selectivity
- Selectivity = FULLKEYCARD/CARD (unique indexes have SELECTIVITY of 1)
 - If table has no index an index gets generated !
 - Avoid this to happen! Add an index before move
 - Generated index use as much non-LOB columns as fit in index
 - Columns with highest number of distinct values first
 - This generates a "fat" index

General Concept – ALLOW_READ_ACCESS

Protocol table SYSTOOLS.ADMIN_MOVE_TABLE

tabschema	tabname	key	value
SAPR3	UTAB	INDEX_SCHEMA	SAPR3
...			



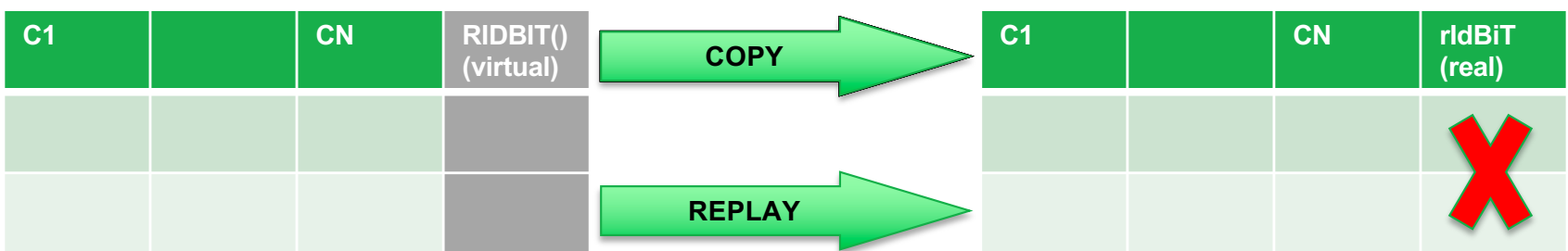


ADMIN_MOVE_TABLE columnar online move

General concept

Protocol table SYSTOOLS.ADMIN_MOVE_TABLE

tabschema	tabname	key	value
SAPR3	UTAB	INDEX_SCHEMA	SAPR3
...			



SOURCE TABLE

TARGET TABLE

CLEANUP calls DROP COLUMN

INIT

- creates target table with additional rldBiT column VARCHAR(16) FOR BIT DATA
- No triggers required. Add TCB flag "MOVE IN PROGRESS"

COPY

- Indexes would slow down copy processing.

REPLAY

- No index required
- Joins source table columns on RIDBIT() with target table on rldBiT column. REPLAY uses DELETE and INSERT (no UPDATE).

SWAP

- Like for row-based tables

CLEANUP

- drops rldBiT column. DROP COLUMN (new in v12.1) requires Z-lock on table
- KEEP option: remove TCB flag, otherwise DROP original table

Stages



The ADMIN_MOVE_TABLE operations can be divided in 5 stages.

- INIT: create staging table, target table and triggers, add TCB “move-in-progress” flag
- COPY: copy data from source table to target table
- REPLAY: apply changes captured in staging table to target table
- SWAP:
 - Online phase: check indexes, create statistics*, ... and replay
 - Offline phase: 1 transaction under Z-lock: final replay, drop triggers, rename tables and indexes
- CLEANUP: drop staging table and original table, KEEP: remove TCB flag, otherwise DROP original table

* only if missing. Statistics creation can be suppressed.

Critical phases



- ADMIN_MOVE_TABLE has its own locking mechanism that avoids that multiple ADMIN_MOVE_TABLE access the system catalogue concurrently. Nevertheless, ADMIN_MOVE_TABLE can collide with other DDL performed on the system catalogue.
- **INIT** phase. After creating the triggers ADMIN_MOVE_TABLE need to be sure that no INSERT/UPDATE/DELETE is running that does not use the triggers.
 - a) Get S-LOCK on table and release it
 - b) Call FLUSH PACKAGES ... HARD INVALIDATION ... for <table> for INSERT,UPDATE,DELETE that all packages get invalidated that access the table

All static SQL gets hard invalidated.

This can cause lock waits for ADMIN_MOVE_TABLE and in concurrent applications.

ADMIN_MOVE_TABLE gives up if lock waits exceed 30 seconds and let other applications pass and then retries multiple times.

- **SWAP** phase (offline part).

ADMIN_MOVE_TABLE tries to acquire Z-lock. 30 seconds timeout + multiple retries (default: 10).

ADMIN_MOVE_TABLE is marked as deadlock looser.

RENAME TABLE/INDEX, DROP TRIGGER perform hard invalidation.

For static SQL applications this means that they might deadlock with ADMIN_MOVE_TABLE during re-validation. ADMIN_MOVE_TABLE will lose, but will also not be able to complete SWAP. The effect for dynamic SQL is similar. New statements cannot prepared while the hard validation is going on and also might deadlock with ADMIN_MOVE_TABLE. ADMIN_MOVE_TABLE will lose, but will also not be able to complete SWAP.
- All this is a consequence of ADMIN_MOVE_TABLE is not REORG, which can easily swap the objects at the end of REORG.

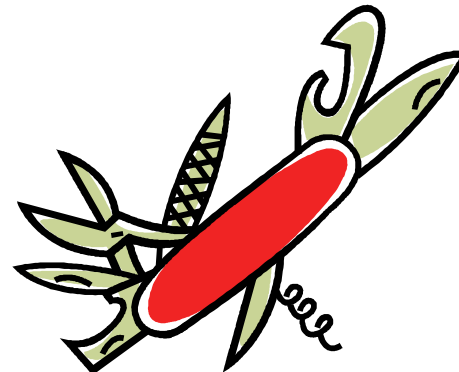
Continuation after ADMIN_MOVE_TABLE or database got stopped



- Use ADMIN_MOVE_TABLE REPORT to list the STATUS
 - STATUS=CLEANUP. Call ADMIN_MOVE_TABLE with CLEANUP option to complete the move.
 - STATUS=COMPLETE: Done
 - STATUS<>CLEANUP nor COMPLETE
 - Use ADMIN_MOVE_TABLE **CANCEL** to revert all changes (drop staging, target, triggers,...)
 - STATUS=INIT: Call ADMIN_MOVE_TABLE CANCEL to revert all changes
 - STATUS=COPY: Call ADMIN_MOVE_TABLE with COPY,REPLAY,SWAP to continue the move
 - STATUS=REPLAY: Call ADMIN_MOVE_TABLE with REPLAY,SWAP to continue the move
- After stopping ADMIN_MOVE_TABLE with TERM option, you need to go through the steps above as well. Check STATUS and decide how you want to continue.

Advantages of this concept

- Online Movement
 - Moving tables to different table spaces with different layout
- Online REORG
 - Online (re-)compression
 - Online REORG even for LOBs
- Online Schema Evolution
 - Online redistribution, change of distribution key
 - Online data type changes
 - Change of table type (to and from)
 - Range-partitioning
 - Multi-Dimensional Clustered (MDC)
 - Insert Time Clustered (ITC)



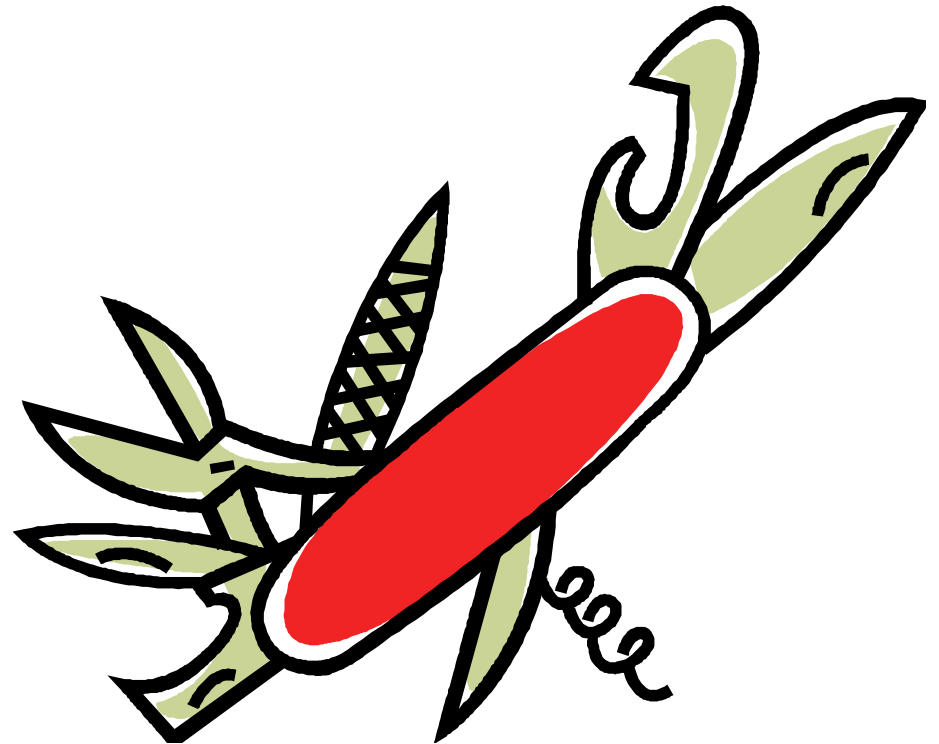
The other side of the coin

“Online” and flexibility does not come for free !

- Extra space required for target and staging table
- Data movement is fully logged and therefore requires additional space in archiving log files.
- Avoiding lock problems cause a performance penalty (e.g. regular commits are required)
- No parallelism, which on the other side is good for low impact for other workload.
- UPDATE, DELETE, INSERT operations get slower on source table due to trigger overhead.



Use cases





Online Move

- Specify target tablespaces
 - SMS → DMS
 - As of 10.1 there is automatic storage
 - DMS → move out LONG/LOB, index
 - Change page size, extent size
 - Move to new tablespace type for example reclaimable space tablespace (V9.7 feature)
- You can use this to online redistribute as well
 - create target tablespaces in target partition group
 - Move table to new partition group
 - Tip: move all tables that belong to new partition group and then call all SWAP operation en-bloc.

Online Schema Evolution - Non-target Table Mode

ADMIN_MOVE_TABLE creates target table with the specified modifications

- Target Tablespace
- MDC/ITC (ORGANIZE BY)
- Range-partitioned (RANGE)
- Change of partitioning key (DISTRIBUTE BY HASH)
- Provide new column definition (e.g. LONG VARCHAR to CLOB conversion).

Note:

- New data type need to be compatible according to CLI rules.
- This can be used to online reorder and drop columns.

```
CALL ADMIN_MOVE_TABLE(<tabschema>,<tabname>,  
  <data_tbsp>,<index_tbsp>,<lob_tbsp>, -- target tablespaces  
  <organize_by_clause>,                -- Sample: ORGANIZE BY INSERT TIME  
  <partkey_cols>,                      -- Distribution key. Sample: 'C1,C3'  
  <data_part>,                        -- Sample: '(C1)(STARTING FROM(1) EXCLUSIVE ENDING AT(1000) EVERY(100))'  
  <coldef>,                          -- New column definition. Sample: 'C1 INT, C2 INT DEFAULT 0'  
  <options>,<operation>)
```

Online Schema Evolution - Target Table Mode

- Tip: Use db2look output of table to be moved as a template

```
db2look -d <database alias> -z <source schema>  
-t <table name> -e -m -o <table name>.ddl
```

- Create target table

```
CREATE TABLE <source schema>.<target table name> (...)
```

- Use target table mode version

```
CALL ADMIN_MOVE_TABLE('<source schema>', '<source name>', '<target  
table name>', '<options>', 'MOVE')
```

Note: Expert feature ! Any conversion that does not cause error is tolerated !

Single versus Multi-step move

Single Step

- Call operation MOVE:

```
CALL ADMIN_MOVE_TABLE('SAPR3',  
    '/BIC/FZSNU28D', 'T2', '', 'MOVE')
```

- Advantage:
 - Easy to use

Multi-step

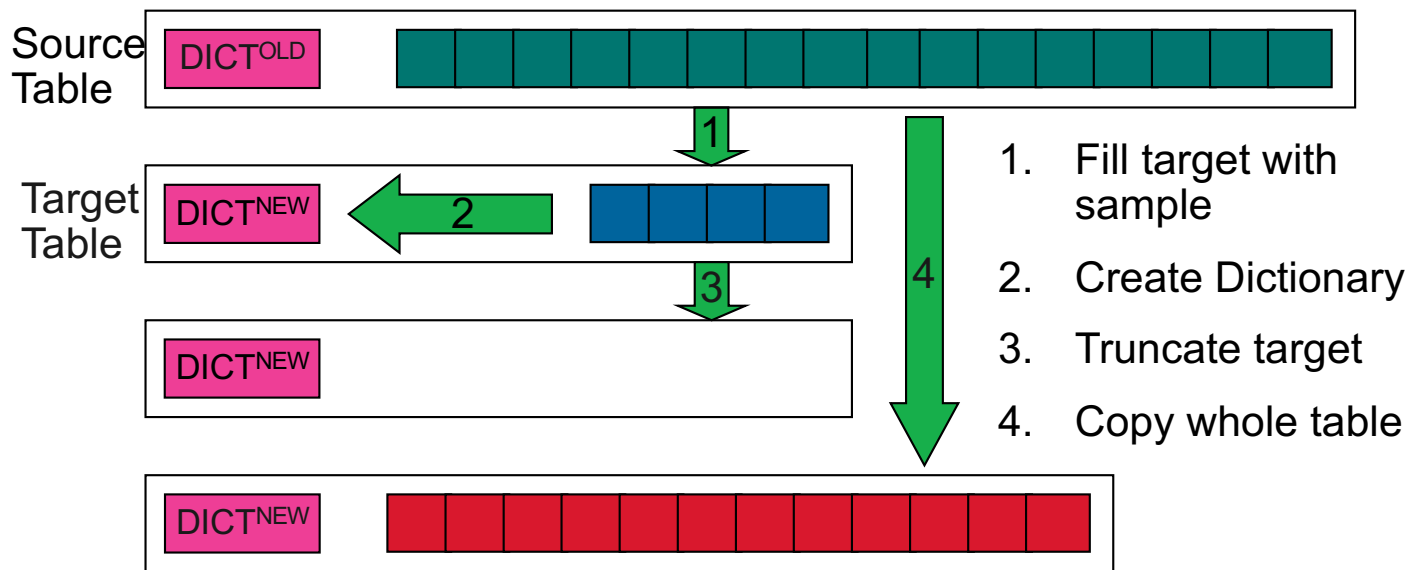
- Call operations separate

```
CALL ADMIN_MOVE_TABLE('SAPR3',  
    '/BIC/FZSNU28D', 'T2', '', 'INIT')  
CALL ADMIN_MOVE_TABLE('SAPR3',  
    '/BIC/FZSNU28D', '', '', 'COPY')  
CALL ADMIN_MOVE_TABLE('SAPR3',  
    '/BIC/FZSNU28D', '', '', 'REPLAY')  
CALL ADMIN_MOVE_TABLE('SAPR3',  
    '/BIC/FZSNU28D', '', '', 'SWAP')
```

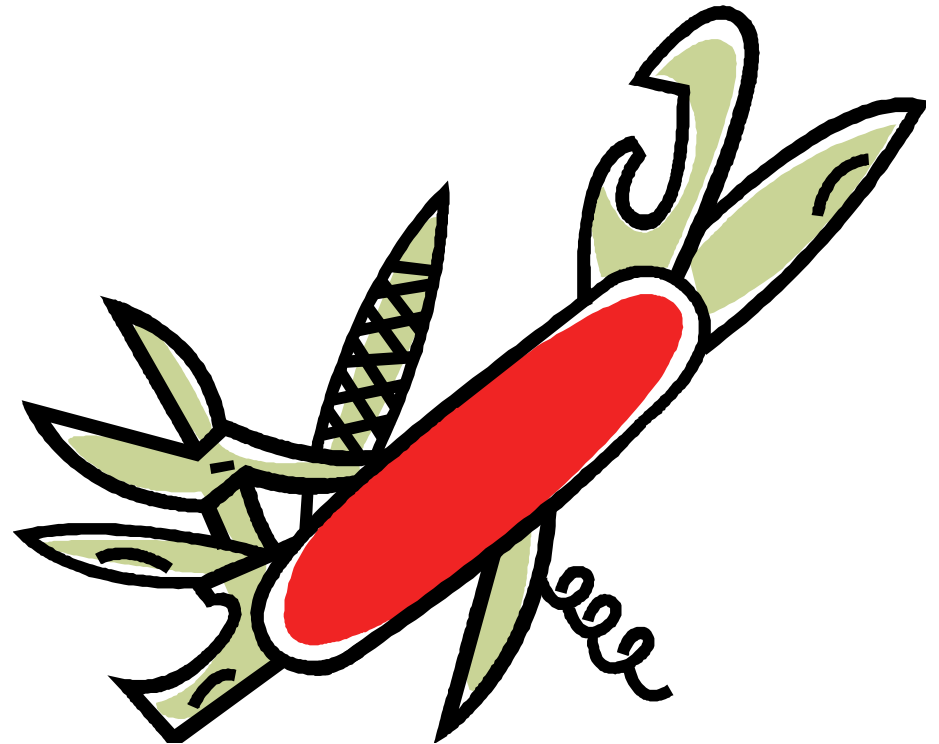
- Advantage:
 - Full control over critical phases INIT and SWAP
 - INIT, SWAP should be performed during times with low system activity.

Online Recompression

- Why using ADMIN_MOVE_TABLE for recompression ?
 - ADC (Automatic Dictionary Creation) lets rows at beginning of table uncompressed + suboptimal dictionary
 - REORG RESET DICTIONARY is offline
- ADMIN_MOVE_TABLE does not rely on ADC. How does it work ?



Features



COPY_WITH_INDEXES (row-based)

- Without indexes (default)
 - Replay index gets created before COPY
 - Other indexes after COPY
 - Advantages:
 - The other indexes are build with "CREATE INDEX" and look optimal before REPLAY starts
 - Disadvantages:
 - CREATE INDEX is a single transaction
 - All indexes need to be created sequentially. Db2 has no DDL to perform the index creation in parallel.
- With indexes (COPY_WITH_INDEXES)
 - All indexes get created before COPY
 - Advantages:
 - No problem with active log space especially on HADR systems with LOGINDEXBUILD=ON
 - Disadvantages:
 - COPY gets slower as all indexes get maintained and log volume is high.
 - All indexes are "used" after the move

LOAD support

Non-recoverable LOAD

- Non-recoverable. Don't use!

Recoverable LOAD

- You need to specify a save location for the **LOAD copy image** of the table that gets created during the LOAD and keep it until the next backup to stay recoverable all the time.

```
CALL ADMIN_MOVE_TABLE ('<schema>' , '<source>' , '<target>' ,  
'COPY_USE_LOAD "COPY YES USE TSM"' ,  
'MOVE' )
```

Performance

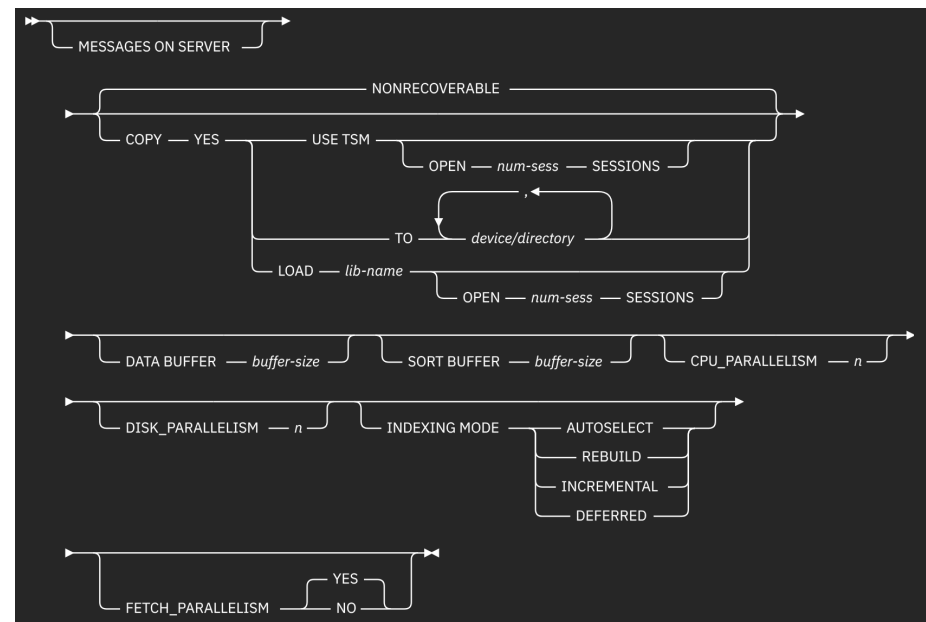
- LOAD is faster
- LOAD is 1 transaction which can lead to active log space full, if LOAD takes long and other transactions fill up active log space.
- The performance overhead for recoverable LOAD is quite small to non-recoverable LOAD (typically below 10%*) !

*Based on IBM internal measurement.

LOAD options supported by ADMIN_MOVE_TABLE

Tuning of the LOAD issued by ADMIN_MOVE_TABLE is possible. You can change the following LOAD options:

- DATA BUFFER
- SORT BUFFER
- CPU_PARALLELISM
- DISK_PARALLELISM
- INDEXING MODE
- FETCH_PARALLELISM



COPY_USE_LOAD option is HADR aware

- If LOAD gets used non-recoverable, HADR standby gets bad.
- If HADR standby cannot apply LOAD copy image, the tablespace goes into recovery pending.
- ADMIN_MOVE_TABLE waits after LOAD until HADR standby confirms that LOAD got applied on standby.
- ADMIN_MOVE_TABLE fails with new error code for long wait time.

REDIRECT

Motivation

- REPLAY takes a long time because the work is done in small chunks to avoid deadlocks.
- If you have high workload REPLAY might not be able to keep up with changes on source table and the move would never end.

Solution

- Allow the redirection of the triggers directly to the target table after the COPY phase is complete

```
CALL ADMIN_MOVE_TABLE('<schema>','<source>','','','REDIRECT')
```

Why can you only switch after COPY ?

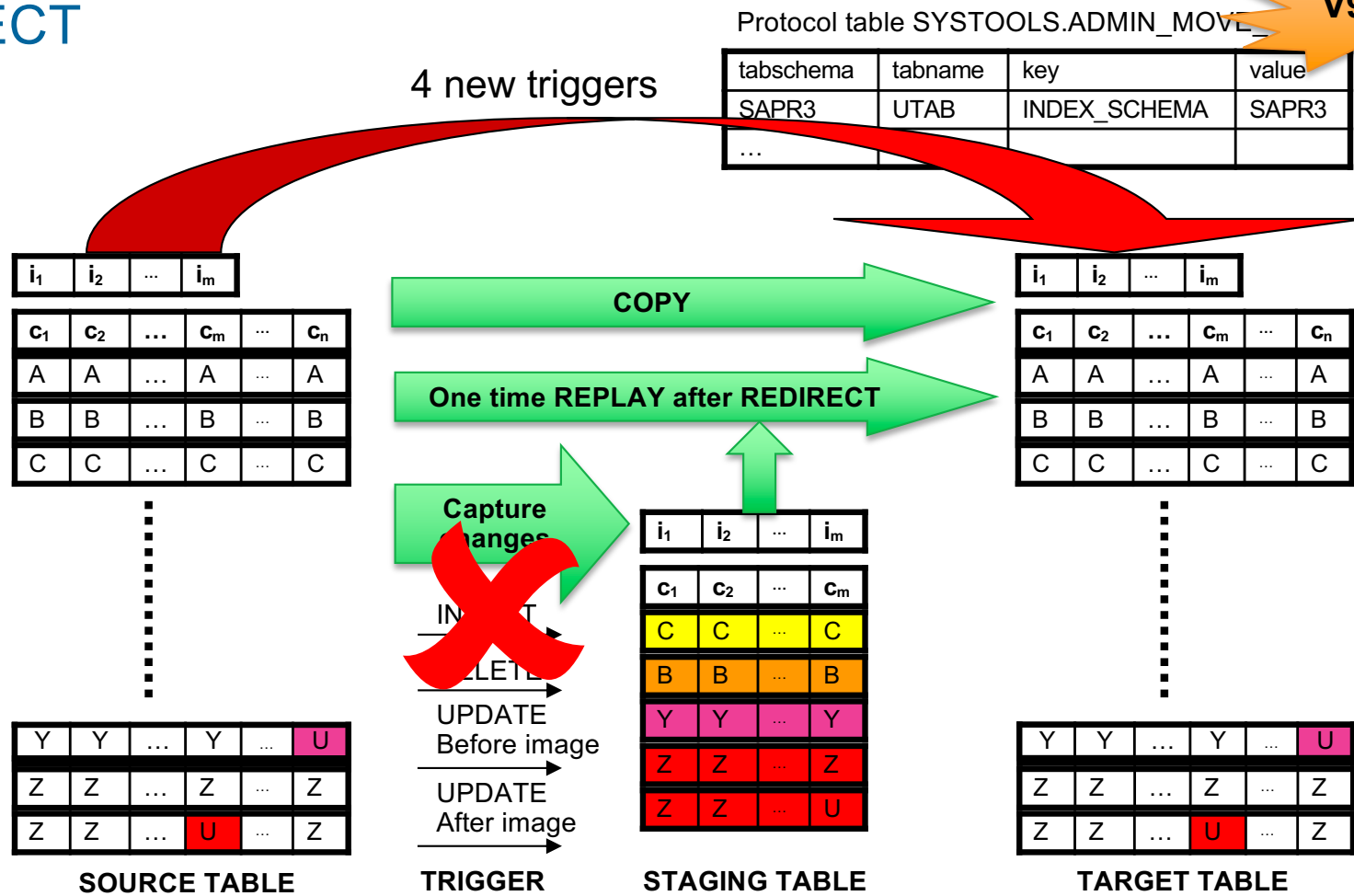
- COPY optimizations would not be possible (LOAD, array insert, block commit,...)

Why is REDIRECT not done automatically under the cover after COPY ?

- Trigger overhead is bigger (use REVERT option to undo REDIRECT, if necessary)
- Not applicable to tables
 - with multiple unique constraints
 - distributed tables that have no unique index

REDIRECT

V9.7 FP5



REPORT



v11.1 GA

Shows statistical information about the move or all moves in progress

Key features:

- Calculates **Estimated Completion Time (ECT)** for COPY and REPLAY
- Calculates throughput numbers for COPY and REPLAY
- Performs LOAD monitoring (if COPY_USE_LOAD used)

Overview of all running moves:

```
CALL ADMIN_MOVE_TABLE('',' ',' ',' ','REPORT')
```

Detailed output for single move:

```
CALL ADMIN_MOVE_TABLE('<tabschema>','<tabname>',' ',' ','REPORT')
```

REPORT overview

**v11.1 GA**

Column Name	Description
STATUS	Status of the Admin Move Table task.
AGENT_ID	Agent Id of a running Admin Move Table task.
INIT_START	Start time of INIT operation of the Admin Move Table task.
COPY_THROUGHPUT	Displays the COPY performance in rows per second.
COPY_ECT	Displays the estimated COPY completion time.
ROWS_STAGING	Displays the number of rows in the staging table.
REPLAY_THROUGHPUT	Displays the REPLAY performance in rows per second.
INFLOW_STAGING	Displays the rate, the staging table increases in rows per second.
OUTFLOW_STAGING	Displays the rate, the staging table decreases in rows per second.
GROWTH_STAGING	Displays the rate, the staging table grow in rows per second.
REPLAY_ECT	Displays the estimated REPLAY completion time.

REPORT single table

**v11.1 GA**

Column Name	Description
TABSCHEMA	Table schema of the Admin Move Table task to be reported.
TABNAME	Table name of the Admin Move Table task to be reported.
STATUS	Status of the Admin Move Table task.
AGENT_ID	Agent Id of a running Admin Move Table task.
COPY_ECT	Displays the estimated COPY completion time.
GROWTH_STAGING	Displays the rate, the staging table grow in rows per second.
REPLAY_ECT	Displays the estimated REPLAY completion time.

TERM

- If you want to stop a running ADMIN_MOVE_TABLE use TERM option

```
CALL ADMIN_MOVE_TABLE ( '<tabschema>' , '<tabname>' , ' ' , ' ' , 'TERM' )
```

- This just stops ADMIN_MOVE_TABLE by issuing FORCE APPLICATION
- Then you need to decide how to continue (CANCEL,COPY,REPLAY,SWAP,CLEANUP) like described earlier.

ESTIMATE

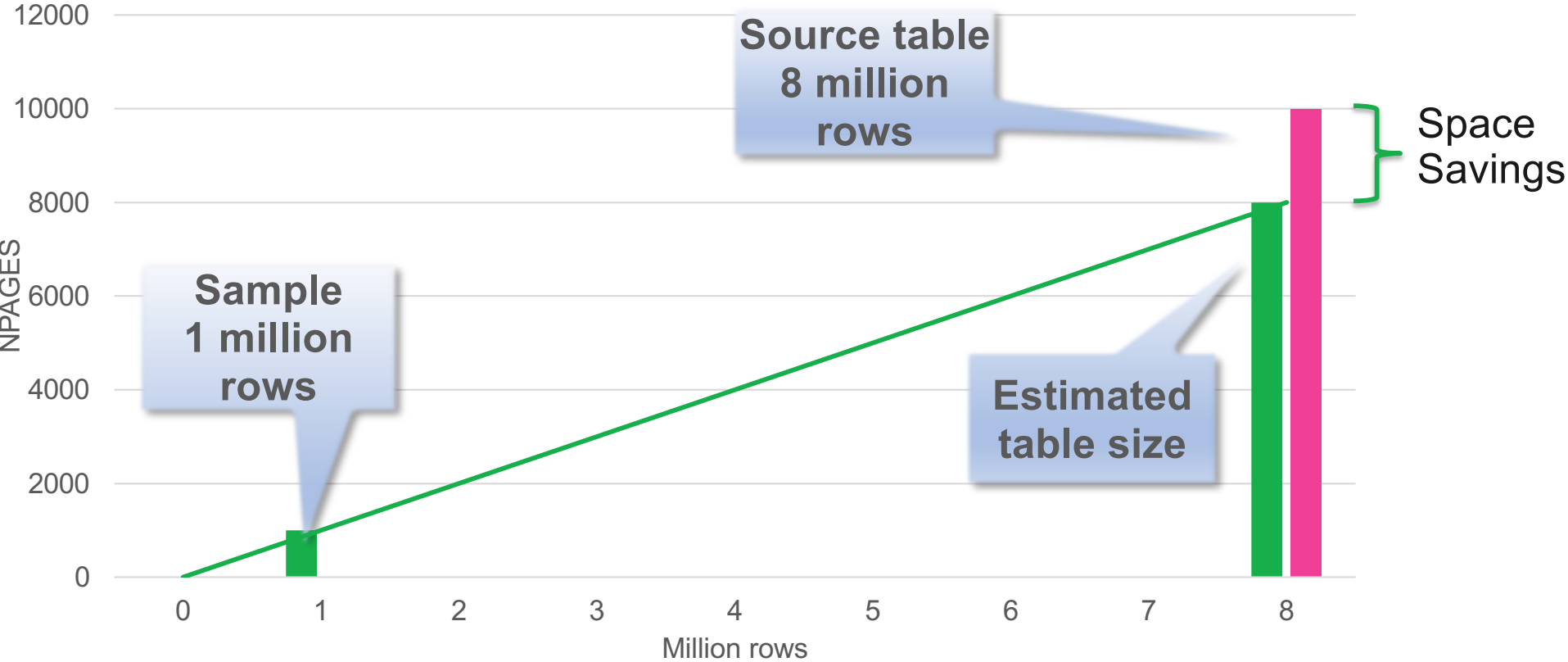


v11.1 M4FP4

- Use case
 - Estimate table size and savings, if table would get re-compressed using optimal dictionary
 - Estimate size before moving from ROW→BLU
 - Estimate size, if moving from BLU→ROW
- Note:
 - Use LOAD to generate dictionary, statistics and loading the sample
 - No locks on source table
 - Can be performed on any table (e.g. does not require indexes)
 - Calculation is based on RUNSTATS information. Requires up-to-date statistics on source table.
 - It also works ROW→ROW, but there are better and cheaper methods exist. Use `ADMIN_GET_TAB_COMPRESS_INFO`
 - Size is an ESTIMATE !
 - Execution is expensive. Do it only on selected tables if values like `PCTENCODED` or `ADMIN_GET_TAB_DICTIONARY_INFO` indicate a potential issue.

ADMIN_MOVE_TABLE ESTIMATE

v11.1 M4FP4

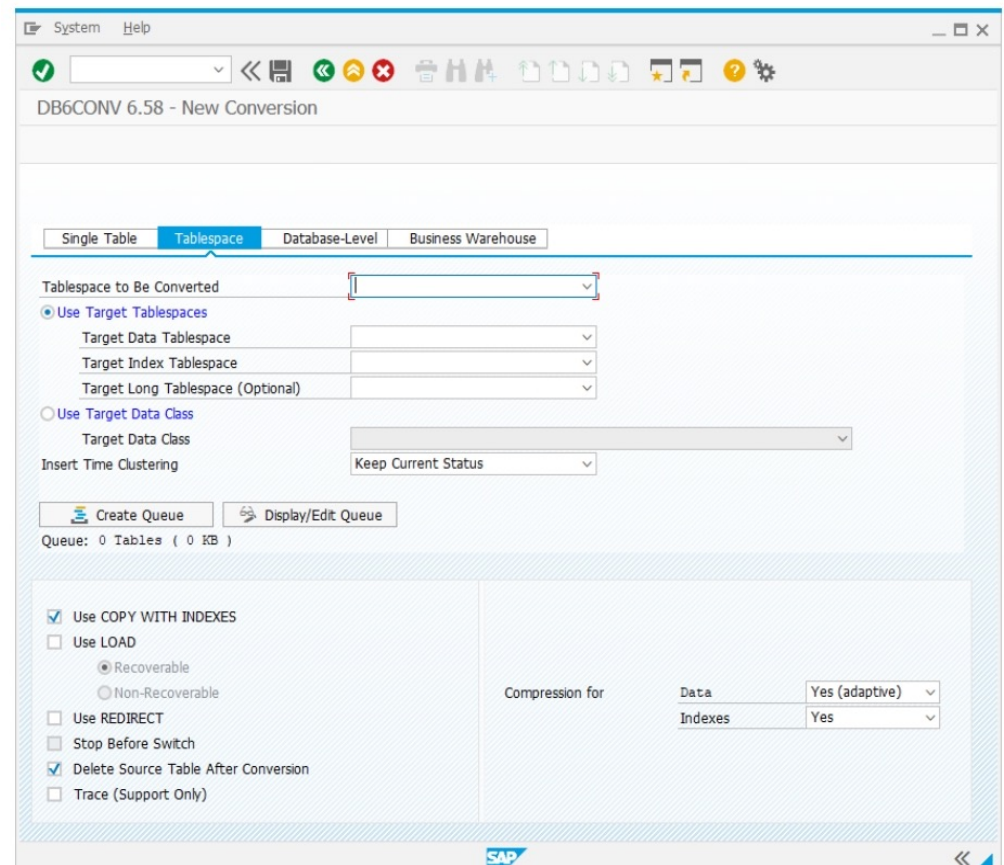


ADMIN_MOVE_TABLE Collect information in db2support as of v12.1

- db2look -m for table, if move is in progress
 - SYSTOOLS.ADMIN_MOVE_TABLE content (global settings and setting per table)
 - AMT REPORT for all tables
 - AMT REPORT for single table
 - ADMIN_GET_TAB_INFO AMT_STATUS
 - Application status of application that suppose to run AMT currently
 - Utility event monitor output if configured (UTIL_START/STOP....)
 - Prints manual cleanup steps (if AMT CANCEL/CLEANUP does not work)
DROP IF EXISTS triggers (4x)
DROP IF EXISTS staging table
DROP target table (if not target table mode)
- Command to clear AMT_STATUS
IN_PROGRESS flag
DELETE statement for protocol table
- Finds inconsistencies between protocol table and reality.
- If table no longer exists
DELETE statement for protocol table
 - Find all tables in AMT_STATUS
IN_PROGRESS. Check if entries in protocol table match.
 - Generate CLEANUP statements if tables "hangs" in CLEANUP phase
 - Checks if application id matches running application. Print all appl_ids mentioned in protocol table

SAP customers

- SAP applications work with dynamic SQL
- ADMIN_MOVE_TABLE is very popular and used as the backend of the SAP transaction DB6CONV
- DB6CONV provides a GUI for ADMIN_MOVE_TABLE and scenarios
 - e.g. DB6CONV can move all tables of a tablespace in one job
- DB6CONV can issue multiple ADMIN_MOVE_TABLE in parallel
- DB6CONV provides an extension for entire SAP BW objects consisting of multiple tables including SAP BW meta data adjustments.



Performance

All numbers provided are based on IBM internal measurement.



COPY versus REPLAY performance*

Table	COPY option	Mio. Rows/h	REPLAY UDI operation	Mio. Rows/h
CDHDR non DPF	INSERT	85	INSERT	3.0
	INSERT with INDEXES	85	DELETE	3.2
	LOAD	115	UPDATE	3.1
DBTABLOG non DPF	INSERT	32	INSERT	3.0
	INSERT with INDEXES	30	DELETE	2.9
	LOAD	120	UPDATE	2.9
/BIC/APSCA000000 DPF	INSERT	46	INSERT	1.3
	INSERT with INDEXES	48	DELETE	1.3
	LOAD	131	UPDATE	1.3
/BIC/APSCA000000 non DPF	INSERT	33	INSERT	1.3
	INSERT with INDEXES	30	DELETE	1.3
	LOAD	62	UPDATE	1.3
/BIC/B0002978000 DPF	INSERT	320	INSERT	1.3
	INSERT with INDEXES	315	DELETE	1.3
	LOAD	467	UPDATE	1.3
/BIC/B0002978000 non DPF	INSERT	109	INSERT	3.1
	INSERT with INDEXES	111	DELETE	3.1
	LOAD	135	UPDATE	3.0

*Based on IBM internal measurement.

COPY_USE_LOAD non-recoverable versus recoverable (to disk)*

COPY_USE_LOAD Throughput non-DPF

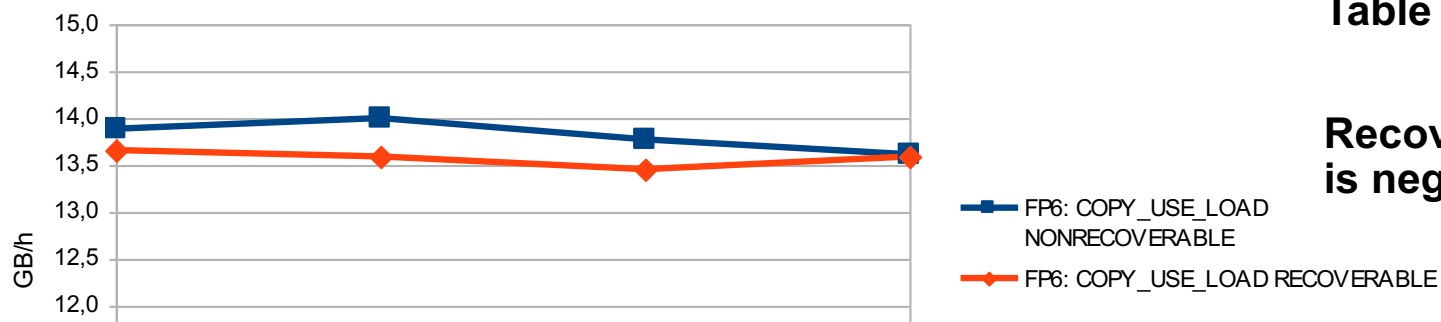
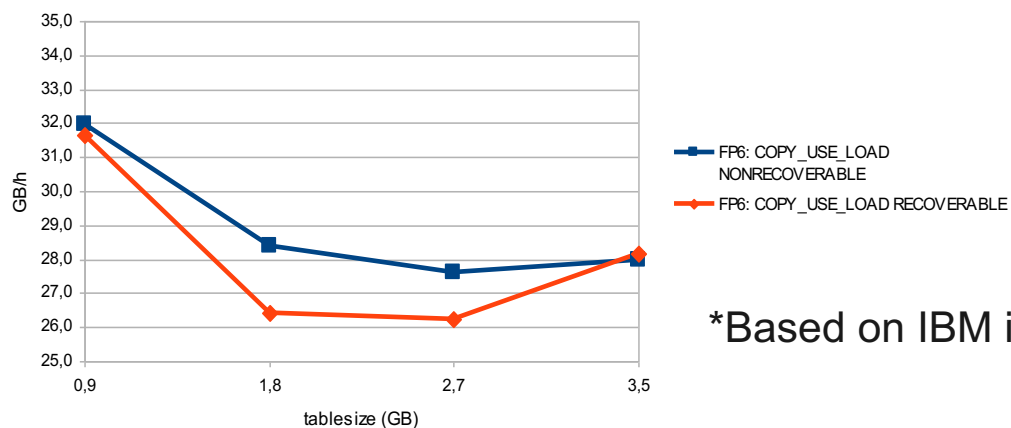


Table /BIC/APSCA000000

Recoverable LOAD overhead is negligible !

COPY_USE_LOAD Throughput DPF



*Based on IBM internal measurement.

Trigger overhead*

UDI operation Table CDHDR	Triggers to staging table	Triggers to target table (REDIRECT)
Single update	82%	120%
Single delete	76%	84%
Single insert	36%	53%

- Triggers slow down UDI operations
- Don't start too many moves in parallel to avoid overloading your system
- REDIRECT triggers are more expensive
- Effects on mass deletes/updates even bigger

*Based on IBM internal measurement.

Known limitations

1. Don't issue TRUNCATE, IMPORT REPLACE, LOAD while move is in progress.
2. Db2ckupgrade will fail if there is ongoing AMT – all ongoing AMTs have to be continued or cancelled before running an Db2 upgrade.
3. REPLAY may not be able to catch up on tables with extreme high number of UDI operations. (REDIRECT)
4. Row-organized tables that have only LONG, LOB or XML columns cannot be moved (→ Exception: ALLOW_READ_ACCESS)
5. If table has only non-unique indexes with bad selectivity, the REPLAY may become a burden.
6. Row-organized tables without index can be moved, but an artificial index gets created on source and target table during the move.
7. Only simple tables are supported as the source table. No materialized query tables, typed tables, range clustered tables, system tables, views, nicknames, or aliases are permitted.
8. A table cannot be moved if an event monitor is currently active on the table.
9. Tables without a unique index are subject to a complex and potentially expensive replay phase.
10. Tables cannot have MQTs on top – RENAME TABLE does not work on tables that have MQTs

Key Takeaways

Remember these 5 things

1. ADMIN_MOVE_TABLE enables table operations while database stays online

Tasks that traditionally required outages can now be done online

2. Start simple, then go advanced

Test with single-step (MOVE) on small tables, use multi-step for production

3. Monitor the REPLAY phase (with REPORT option)

That's where problems usually show up

4. Plan your timing carefully

INIT and SWAP need low activity, COPY can run anytime

5. It's not magic, it's engineering

Takes time, has overhead, but keeps your database online

Getting Started

Your next steps

- ☐ **Identify a test table in non-production**
Start with something small and non-critical
- ☐ **Try a simple move to different tablespace**
Use single-step MOVE operation
- ☐ **Monitor it using REPORT command**
Watch COPY and REPLAY phases
- ☐ **Review the documentation**
Db2 Knowledge Center and SAP Notes
- ☐ **Plan your first production move**
Use multi-step for control

References

- For SAP Customers:
 - [SAP Note 1513862](#) - DB6: Table conversion using DB6CONV version 6 or higher
- Db2 Knowledge Center:
 - [ADMIN MOVE TABLE procedure – Moving tables online](#)
 - [Moving table online by using ADMIN MOVE TABLE](#)
- Technotes
 - [What to do if the CANCEL operation of ADMIN MOVE TABLE fails](#)

Questions?

Beck Tang - becktang@ca.ibm.com

Jens Seifert - Jens.Seifert@de.ibm.com

Phil King - Phil.Leslie.King@ibm.com

Martin Jungfer - martin.jungfer@de.ibm.com

Zahidul Khan - zakhan@ca.ibm.com



Call for Sessions

Submission deadline: [May 22](#)

IBM TechXchange 2026

October 26–29 | Atlanta, GA

Clients, Partners & IBMers are eligible to submit sessions

Share your expertise with thousands of technical professionals!

What we are looking for

- Real-life use cases and implementation stories
- Technical deep dives (non-promotional)
- Demonstrated expertise with IBM products
- Sessions aligned to 2026 priorities: enterprise AI applications, key products, technical skills
- A compelling title and concise, clear abstract

Session types



Technology Breakout (45 minutes)

In-depth technical session featuring architectures, demos, best practices, client/partner stories



Tech Talk (20 minutes)

Fast-paced spotlight on use cases, technical trends, and special-topic insights.

Client & partner submissions only for Tech Talks.

Technology areas

AI Applications | Data Management

Cloud & Platform Engineering | Infrastructure

Security, Risk & Compliance

Special perks

- A complimentary full conference pass (that's a \$1599 USD value!)
- Speaker training and content support
- Social tools to promote your session

Submission deadline: [May 22](#)

Get started



© 2026 IBM Corporation

Thank you

