
CDB/Auto-Load

CDB/Auto-Load is a program that loads data from sequential files into DB2 tables and their associated indexes in a fraction of the time it takes the IBM DB2 LOAD utility to do the same function. The primary objectives of CDB/Auto-Load are reliability and speed. For LOAD RESUME, a third primary objective is data availability.

This chapter summarizes the main features and capabilities of CDB/Auto-Load.

CDB/Auto-Load Features

Two Modes of Operation

For maximum flexibility, CDB/Auto-Load can function in two different modes: A two-phase mode for high parallelism, and a passthrough mode that provides the best performance for non-partitioned tablespaces.

Two-Phase Mode

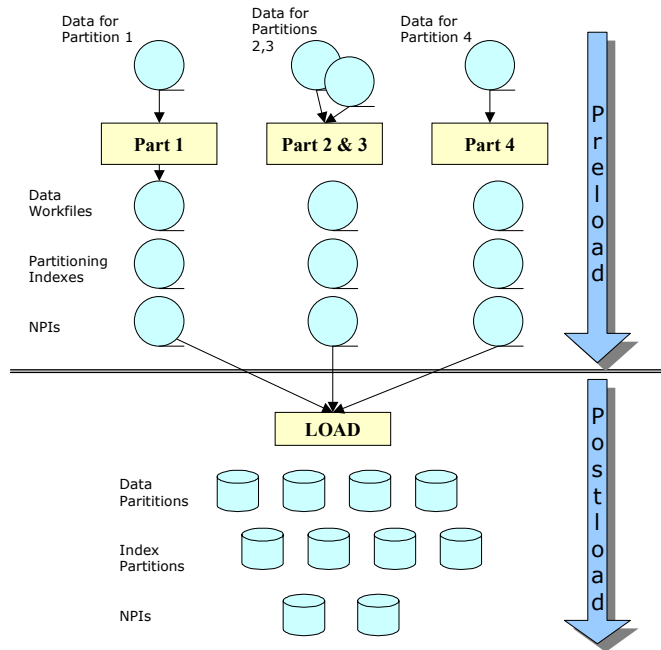
When running in two-phase mode, CDB/Auto-Load executes two distinct processes:

- **PRELOAD:** Reads input data and writes DB2 data rows and index entries keys (for both the partitioning index and any non-partitioning indexes in the case of partitioned tablespaces) to sequential datasets.
- **POSTLOAD:** Reads the output datasets from the Preload phase and Buildings DB2 spaces (tablespace and indexes) from the output of the first process.

These processes are shown graphically in figure “*The 2-Phase Load Process.*” Note that, if there are multiple input datasets and they are sorted, the Preload phase can process them in parallel, greatly reducing the elapsed time required to complete the phase. The Postload phase will also process multiple data and index partitions in parallel.

You have the option the run **Preload**, the first process, separately from the second process, the actual **Load**, or run them together in a manner similar to the IBM LOAD utility.

The majority of the *work* that goes on during the Load process is in the Preload phase. You can therefore implement a process by which you run Preload and



The 2-Phase Load Process

Load at different times. This could have several benefits.

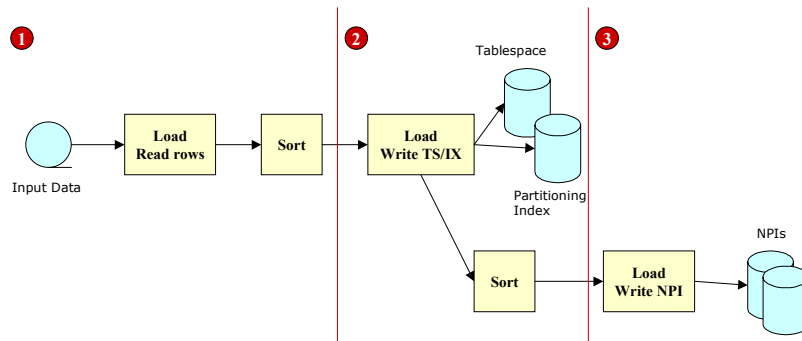
- Verify that the input data to be loaded is valid.
- Allow applications to continue to run while the data for Load is being prepared.
- Minimize the unavailability of your data and indexes while the very fast portion of Load, the LOAD phase, runs.

The structure of CDB/Auto-Load permits you to minimize the outage of DB2 objects due to utility functions that must operate within a constrained time period or batch window.

When running LOAD REPLACE, you do not have to run these processes separately; they can be run in a single step.

Passthrough Mode

You have the option to instruct CDB/Auto-Load to use a fast technique called PASSTHRU. The feature merges the Preload and Postload phases into a single phase and does not use intermediate sequential datasets to hold keys or data. See the figure “*Load Passthrough Mode*” for details about this mode. CDB/Auto-Load reads rows from the input dataset and passes them to sort (1 in the



Load Passthrough Mode

figure). Once sort ends, CDB/Auto-Load reads from sort, and writes the data to the tablespace and the index. If there are non-partitioning indexes (NPIs), then data is passed to sort once more, this time to process the NPI keys (2). After the termination of the sort, CDB/Auto-Load writes the NPIs (3).

Optimized Dataset Allocations

Tablespace allocations are automatically optimized by CDB/Auto-Load based on the size of the data to load. No complex, time consuming calculations are required.

Optional Sort of Input Data

CDB/Auto-Load will, at your request, invoke SORT for your **input data**. CDB/Auto-Load will generate the SORT based on the definition of the Clustering Index. If no explicit Clustering Index is defined, sort will be skipped for the data. With the IBM LOAD utility, you must perform a separate step prior to Load if you require your data to be in a particular order.

If you elect not to SORT, then CDB/Auto-Load checks the order of the input data against the defined Clustering Index. If any rows are out of sequence, and discarding is not active, the Load will terminate.

Extensive Data Parsing and Conversions

The following statements are supported by CDB/Auto-Load:

```
NULLIF  
WHEN  
DEFAULTIF  
CONTINUEIF
```

The conversions supported by CDB/Auto-Load are a superset of those provided by IBM load. The intelligent mechanism used by CDB/Auto-Load to recognize input data as numeric and perform any needed conversions is extremely useful in situations where you must now perform separate processes on your input data in order to make it acceptable for the IBM LOAD utility.

Discards are also supported by CDB/Auto-Load.

Cleaning-up Input Data

As an option, CDB/Auto-Load can create a “cleaned-up” version of the input data with duplicate keys and discards removed. This dataset can then be used in other processes, and may save you the need to unload the data after a load.

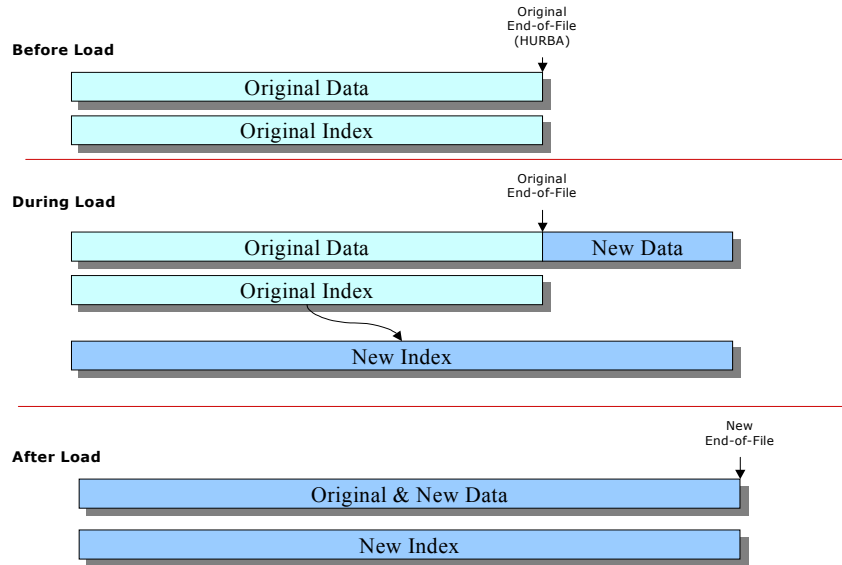
Additionally, CDB/Auto-Load can run in “Check Data” mode. In this mode, the tablespace is not loaded, but the data is checked to determine if it contains any duplicate keys or discards. This feature can be used in conjunction with the previous one in order to produce “clean” data that may speed-up the actual load phase.

Updating RUNSTATS

CDB/Auto-Load can, at your option, update the DB2 catalog with the RUNSTATS values calculated at the end of the Load. This is not available when running LOAD RESUME.

Parallel Processing

CDB/Auto-Load provides you with the ability to process partitioned tablespaces in parallel. You can supply CDB/Auto-Load with input data files for one or more partitions. The **Preload** process requires that the input be on a partition boundary, as determined by the DB2 limitkey values. The number of partitions contained in each input data file is a user choice. You may supply all data in a single file, or you may run Preload separately for each partition or any combination in between. It is therefore possible, if the application that supplies the data to Load is so designed, to run Preload simultaneously for every partition.



Non-Destructive Load Resume

Each of these must be a separate job.

The **Postload**, or **LOAD**, phase always runs all partitions in parallel, but within a single job step.

Concurrent Copy

CDB/Auto-Load can create DB2 Image Copies concurrently with the Postload or **LOAD** process when you run **LOAD REPLACE** on a non-partitioned tablespace. These copies are specified with standard IBM **COPY** Utility statements. These copies are created as part of the Load process; a separate run is not made. In fact, the copies are made concurrently with the loading of the data into the tablespace.

Non-Destructive LOAD RESUME

Often, **LOAD RESUME** is used to load a relatively small portion of a tablespace (maybe 100,000 rows in a 10 million row tablespace, or 1% of the total rows). Non-CDB load approaches often put the entire tablespace at risk; that is, if the job fails for any reason, all the data is destroyed, not just the part being loaded. In contrast, CDB/Auto-Load is non-destructive when running in **LOAD RESUME** mode. If, for any reason, the job fails before completion, existing data is not affected in any way. This eliminates the need to take an image copy prior to the **LOAD**, and avoids the possibility of running lengthy recoveries if some-

thing goes wrong.

The figure entitled “*Non-Destructive Load Resume*” shows how this option works. Before the load starts, the “used” (logical end of file) portion of the tablespace is determined by the VSAM field “High-Used RBA.” During the load, CDB/Auto-Load adds data to the end of the tablespace without moving the High-Used RBA. At the same time, a new index is created which includes the keys from the original index and the new keys. The original index is left unchanged.

If a failure occurs, the index is unchanged, the HURBA has not been moved, and the indexes are left unmodified. Data and indexes are immediately usable. Once the load completes normally, the HURBA is moved to its new location, and the new index replaces the old one.

Load with Tablespace in Read-Only Mode

CDB/Auto-Load lets you perform a LOAD RESUME on your tablespace while it is active to DB2 in READ-ONLY mode. The tablespace is stopped for only a few seconds at the end of the load.

Speed and Low Resource Consumption

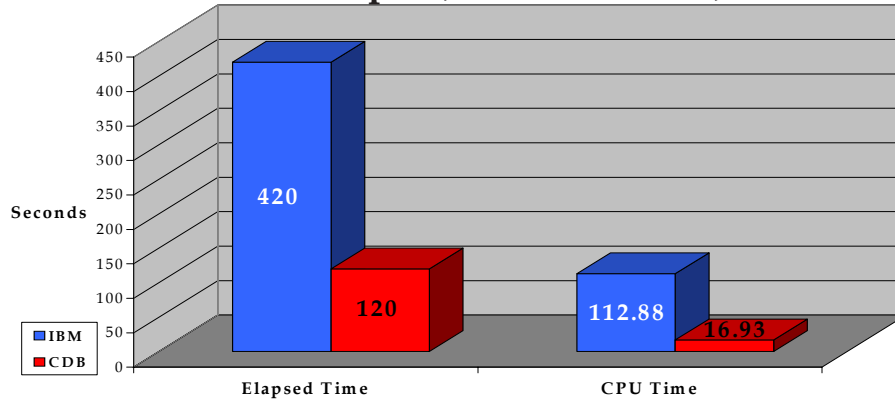
CDB/Auto-Load is significantly faster than its IBM counterpart, and uses far fewer CPU resources.

Compatibility with IBM Load Control Statements

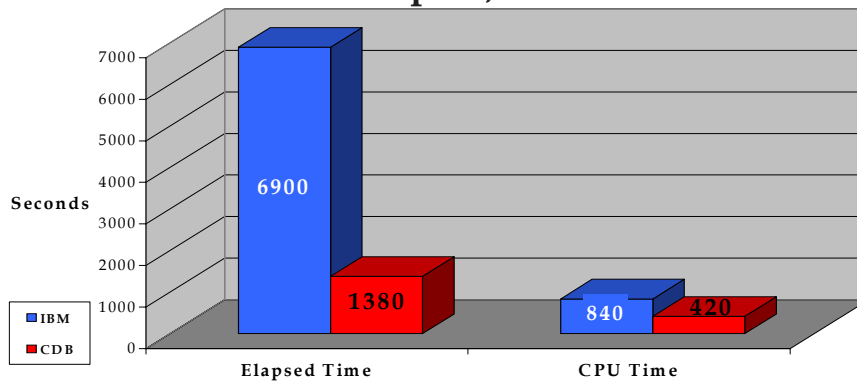
CDB/Auto-Load input statements are compatible with those used by IBM. This greatly simplifies the task of converting to CDB/Auto-Load.

CDB/Auto-Load Benchmarks

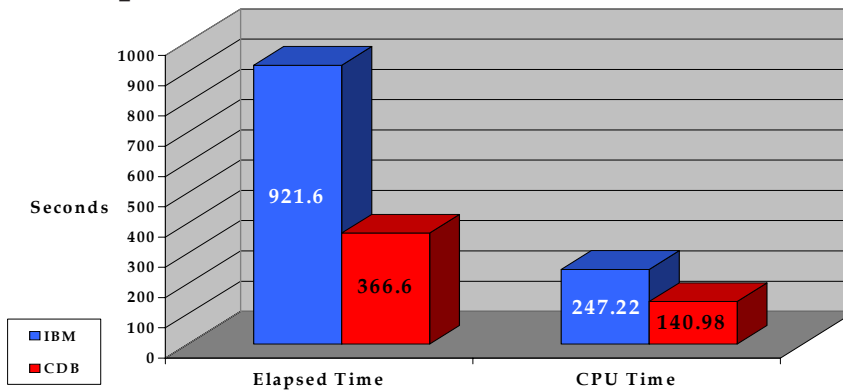
Auto-Load (V7)
Partitioned Tablespace, 2 NPIs and 900,000 Rows



Auto-Load (V7)
Partitioned Tablespace, 20 Million Rows



Auto-Load (V7)
Simple Table, 3 Indexes, and 2.4 Million Rows



CDB/Auto-Load Benchmarks

