

# Midrange Dynamics

## Manage IBM i Database File Changes & Data Transformation – Deploy in Seconds



### Are database modernization projects or file expansions on your horizon?

Critical applications often need to be available 24/7 so downtime must be minimal. It's difficult to tackle projects like application and database modernization without disrupting systems that support vital business operations.

Typically, when a file structure is modified, users lose access to one or more applications while the change is implemented. Issues arise when you can't fit the data copy or conversion

into an available downtime window due to file size. The need to quickly deploy changes to high volume files on multiple servers compounds this challenge.

**MDTransform, MDRapid, and Midrange Dynamics change management (MDCMS)** work together to substantially reduce application downtime and risk as you deploy database changes across your enterprise. Your IT team can respond to business needs and enhancement requests with agility. Moving to modern, efficient database architecture becomes easier to plan and achieve.

### Manage Database File Changes with MDTransform



IBM i development teams modernizing application architecture with SQL, or just modifying existing file structures, will appreciate the data transformation capability within MDCMS. Complex SQL syntax and conversion programs can become a thing of the past with MDTransform. It helps you quickly map and manage database file changes, and it even generates SQL scripts.

Using MDTransform, confidently automate managing field expansions, initializing new fields, merging fields, and managing deployment of DDS to DDL conversions. It also handles field type changes such as numerical to string, string to graphical, and vice versa, with ease.

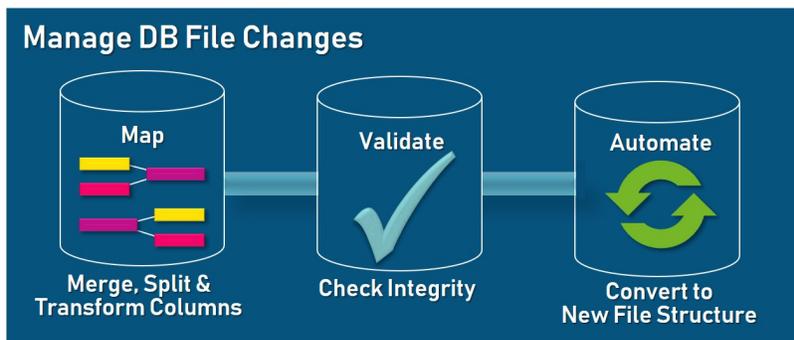
MDTransform guides users through the steps to easily add and populate new fields or update field content. Behind the scenes, it builds and executes a dynamic SQL insert to populate the new version of the file.

Data transformation results are validated during the compile phase to avoid surprises during installation. In fact, users can view, customize, and validate conversion results when transformation occurs. MDTransform can be accessed from MDCMS Object Manager or MDOpen, the Eclipse-based plugin to MDCMS.

### Build and Refresh Test Data

Providing reliable test data is critical for any test plan. MDTransform uses MDCMS Data Copy Templates to identify selection criteria for building and maintaining test data. Select which files, and which records in the file, you need. MDTransform can even initiate column-specific masking of confidential data.

Development and QA teams can quickly refresh data whether it resides on the same or different partitions. Templates may be saved, copied, and reused, and data transfer can occur dynamically or be scheduled for periodic refreshes.



[www.midrangedynamics.com](http://www.midrangedynamics.com)



MIDRANGE DYNAMICS  
providing innovative IBM i solutions



## Reduce Application Downtime and Risk with MDRapid



MDRapid uses MDTransform to convert and migrate data to new file formats while the application is in use. Your business applications continue to access databases as usual while upgrades are taking place. Any changes made to the current version of the files are journaled and synced with the new version held in the staged library. Thanks to a low performance footprint, users are not impacted during this process.

Since MDRapid deploys database changes quickly, even for multi-million- and multi-billion record files that have an extensive amount of update activity, it's easy to integrate MDRapid into day-to-day operations.

For example, consider a table with 388 million records. There are 101 logical files over the table, 30 of which are join files to additional tables:

- Time to transform data into new version of table, including the build of the 101 logicals, using MDRapid while application is still active: **90 minutes\***
- Time for MDCMS to need application in lock state to move in new version of all files: **17 seconds**
- Time a change management product not using MDTransform/MDRapid would need to have application in lock state while performing a CPYF on same machine: **15 hours**

### MD Extras

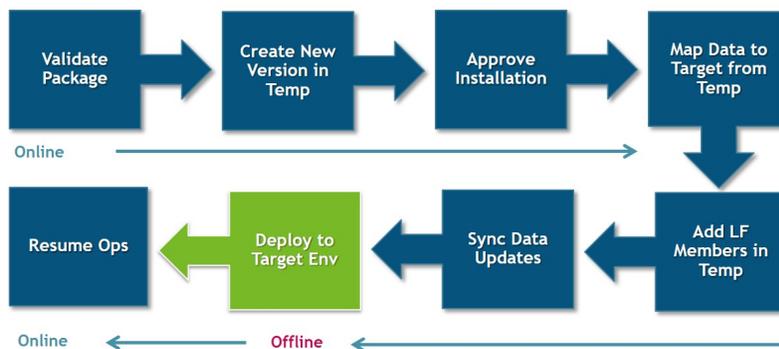
**Helpful job monitor** — Display job status, total number of records, % complete, time estimated to complete, and actual time completed.

**Status Updates** — Convenient email notifications and status exit points keep you informed.

**Automatic installation** — Specify a time window for installation of database changes.

## How MDRapid Works

Taking full advantage of control and auditing functionality inherent with change management, MDCMS and MDRapid work in concert to manage database updates and implementation of all associated application objects.



1. MDCMS creates a new version of the DB2 file or SQL table in a temporary library.
2. MDCMS checks to see if the current version contains sufficient records to require use of MDRapid based on application preferences.
3. User or system

approves the launch of MDRapid.

4. MDRapid creates empty members in the new format of the staged files.
5. It ensures that the limits for record count and member count are high enough to accommodate all records and members in the current version of your production DB2 file or SQL table. It also provides for replicating a multimember DB2 file to several SQL tables.
6. It copies all records from the current version to the new version. You can specify custom data conversion and mapping parameters.
7. It builds any logical files, including views and indexes once the initial copy is complete.
8. It simultaneously migrates any transactions performed in the live application until such time that the deployment of the changes should occur.
9. MDCMS moves all impacted objects to a backup library and moves the new files and related objects into the application. If anything goes wrong, MDCMS's robust installation and rollback capabilities protect your production environment, and you can roll back to your old production data.

