

DATAMATCH: AUTOMATED BATCH TESTING

FACT SHEET



AT A GLANCE

DataMatch is a powerful, scriptable database comparison tool. It enables quasi automatic testing of batch jobs, and can help reduce testing effort dramatically.

DataMatch and its sibling product TestMatch (for automated on-line testing) form the core of the Anubex Automated Testing offering for mainframe migration projects.

DataMatch also works closely together with DataTurn, for automated data conversion.

Anubex mainframe migrations guarantee 100% functional equivalency. This means that running a converted batch job on the new platform should yield exactly the same results as running the original JCL job on mainframe.

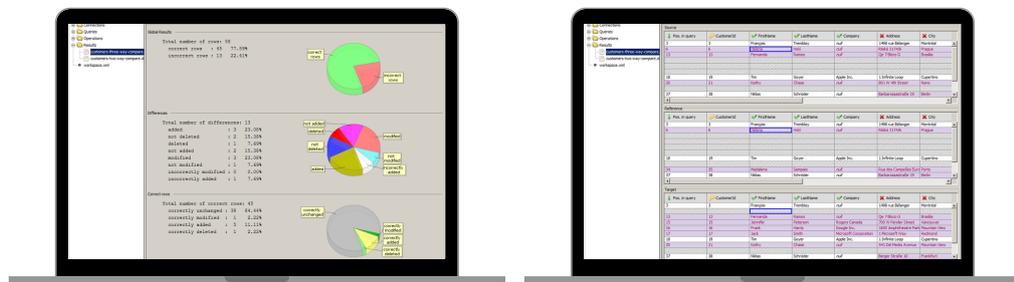
To prove this, DataMatch basically compares the output of a converted batch job (e.g. Perl on Linux or PowerShell on Windows) with the “desired” output i.e. that of the original JCL job on mainframe. While this sounds straight-forward at first, several issues make this a challenging task:

1. How to compare the contents of exotic mainframe databases such as IDMS and Adabas with the converted relational database on the target platform?
2. Data volumes, typically in the Terabyte range, might prohibit “naive” bulk-comparisons.

DataMatch is designed with these challenges in mind. The result is a unique and flexible tool that makes batch testing a breeze in any situation.

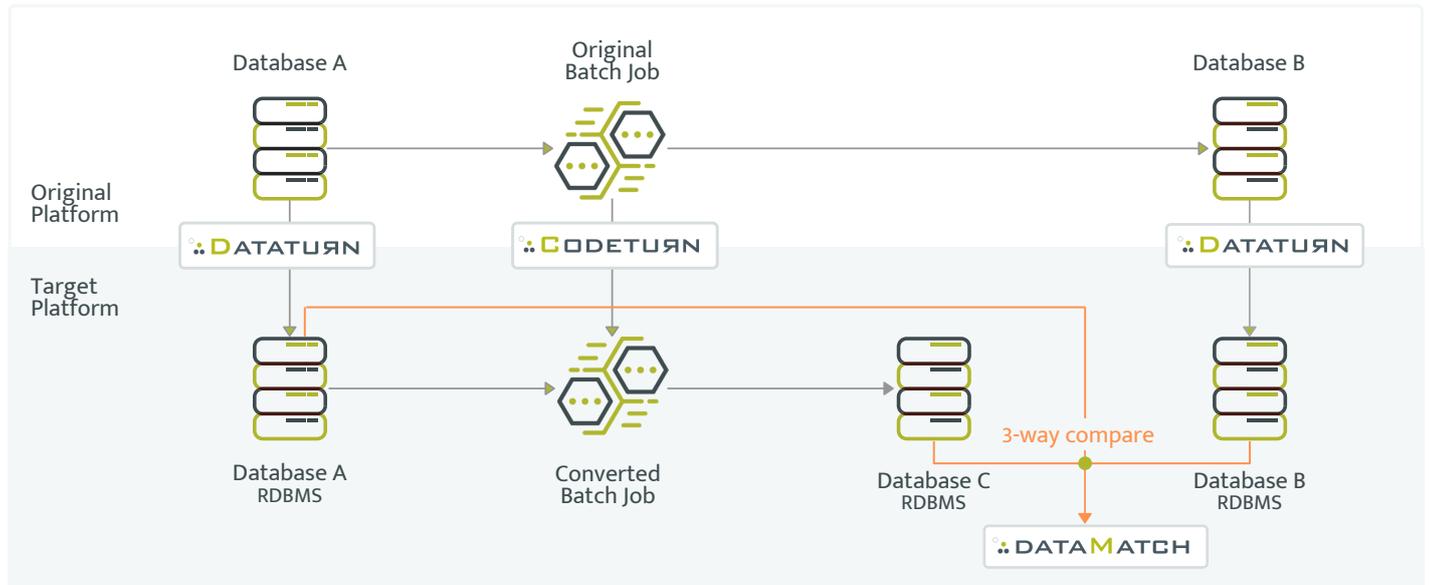
OVERVIEW

DataMatch has two main components:



- A graphical user interface where database connections are set up, comparisons are configured and results are analyzed.
- A scriptable engine that can execute previously configured comparison runs, and that can be easily integrated in other testing frameworks, logging and monitoring tools etc. This way, even long-running batch jobs can be tested without manual intervention.

The DataMatch GUI offers powerful tools to analyze and track down any differences found. It can zoom in on specific differences, or detect patterns in the differences using various correlation techniques.



HOW IT WORKS

The process of migrating and testing a batch job is described in the above diagram. It starts by capturing the initial state of the database (A) on the mainframe and transferring that to the target platform. If the original database is not already relational, it is converted using DataTurn.

Then, both the original job and the converted job are run. The final database state on mainframe (B) is the “desired” result; the final database state on the target platform (C) should be identical.

Before comparison takes place, database (B) is converted to a relational form again if needed.

As a last step, DataMatch performs the actual comparison to effectively prove that both jobs have produced the same result.

These steps can be easily automated so they can be repeated or re-used for subsequent jobs. Moreover, as long as database states A and B are retained, the converted batch job can be re-executed immediately, allowing the test to be repeated quickly and as often as required.

DataMatch internally uses JDBC to access all three databases (A, B, and C) so there is no limitation as to what databases can be compared.

Anubex is an industry leader providing core software solutions, know-how, and support for complex mainframe-to-cloud migration and modernization projects.

Get in touch at migrations@anubex.com

Copyright ©Anubex | www.anubex.com

THREE-WAY COMPARE

While comparing the database states, three types of differences can occur:

1. The same record in (B) and (C) contains different data.
2. A record occurs in (B) and not in (C).
3. A record occurs in (C) and not in (B).

While type 1 are trivial, type 2 and 3 are not: does a missing record in (C) mean that it was not inserted, or that it was deleted by mistake? This can only be found out if we also take the initial database state (A) into account. DataMatch effectively implements this three-way compare.

OTHER USES OF DATAMATCH

DataMatch is most often used in the context of mainframe migrations and particularly in testing of JCL conversion. However it is just as useful in various other situations. For instance:

Regression testing in day-to-day maintenance. Programs that are unaltered should produce the same output before and after the release. DataMatch can be used to visually inspect the result of bulk data operations (e.g. mass cleanup of historical data from tables).

When performing system upgrades (operating system upgrade, database upgrade), DataMatch can validate that programs still work correctly after the upgrade.