

Are you concerned about time, money & resources being spent on Data Testing?

What are your options?

You can do nothing,

Because you don't have the automation solution for ETL testing from iCEDQ or a mature in-house system.

Manual testing can be slow and expensive. Experts writing custom scripts are expensive too. Your global team finds it difficult to collaborate. You have project delivery risk because of quality issues and there is little visibility into the ETL black box.

Or, you can do something!

Get the iCEDQ ETL testing platform and drastically reduce your project time, money, and resources.

iCEDQ is a web based data testing solution that supports team collaboration. Its' rules execution engine is specifically designed for data testing. With iCEDQ you can compare and test data across databases after ETL process execution. It integrates with HP Quality Center as well.



Don't develop your ETL or data centric project without test automation. Get the iCEDQ solution.

To learn more

about how iCEDQ can help you audit and reconcile your data flows, please contact us 1 888 701 3477, email sales@icedq.com or visit: icedq.com

iCEDQ provides testing and development teams of all sizes, the ability to interact and collaboratively develop, use, and manage ETL testing for their data-centric projects.

Today's development projects are overwhelmingly data-centric. Such projects have inherent challenge of ETL testing, ultimately impacting development timelines. Testing solutions must be able to address this challenge, which is not present in traditional projects.

Where will I use iCEDQ? And what will it do for me?

There are many pitfalls in manual or homegrown testing solutions for data-centric projects. These approaches require recruitment of programmers who create custom scripts. Rather than focusing on test cases and testing, testers typically end up spending more time on development.

Are you still writing custom code for ETL testing? Then it is time to STOP!

```
Dim arrRecordsSrc2
Set StatusDictObj_Pass = CreateObject("Scripting.Dictionary")
Set StatusDictObj_Fail = CreateObject("Scripting.Dictionary")

Loadfile()

'ConnStr = 'DRIVER={Sybase}
'ProcName = 'Qa_Load'

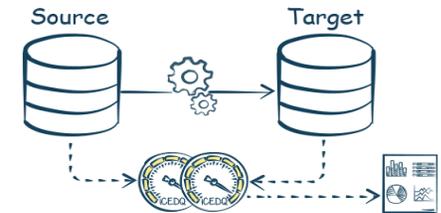
'ConnStr = 'DRIVER={Sybase}
'ConnStr = 'DRIVER={Sybase}
'ProcName = 'Qa_Load'
ExecuteStoredProc ConnStr

DataTable.AddSheet "Source1"
DataTable.ImportSheet Environment.Value("Data_Sheet"), "Expected output", "Source"

DataTable.AddSheet "Mapping"
DataTable.ImportSheet Environment("Data_Sheet"), "Mapping_Products", "Mapping"
```

Such a solution can neither fully integrate with its environment, nor can it be robust.

iCEDQ engine can be used to test ETL/ Data processes that involve source and target databases. Rules such as row-counts, checksum of amounts, and comparison of list of values between source and target can be quickly defined and executed.



How iCEDQ features help?

- ✦ iCEDQ requires no custom code or writing SQL.
 - ✦ Test automation saves developer time.
 - ✦ The built-in scheduler allows scheduling of test to run at night after the ETL batches are executed.
 - ✦ With iCEDQ, validate large dataset instead of subsets, unlike the size limitations in spreadsheets based approach.
 - ✦ With iCEDQ, global teams can collaborate to perform tests, and monitor progress regardless of physical location or time. iCEDQ dashboard provides transparency and metrics for management and project progress. Data-centric projects do not follow rigid SDLC. iCEDQ supports agile or/and iterative processes.
 - ✦ Testers can easily translate business requirements into physical executable rules.
 - ✦ iCEDQ rules can be easily re-pointed to different database environments such as development, QA, UAT.
 - ✦ iCEDQ supports regression testing of existing functionality with new release. Open API allows integration of iCEDQ with tools such as HP Quality Center¹
- At the end of the day it comes down to time, money, and resources.

The ROI for iCEDQ is typically delivered within a few months - not years.