

# erwin<sup>®</sup> Automation Framework

## AT A GLANCE

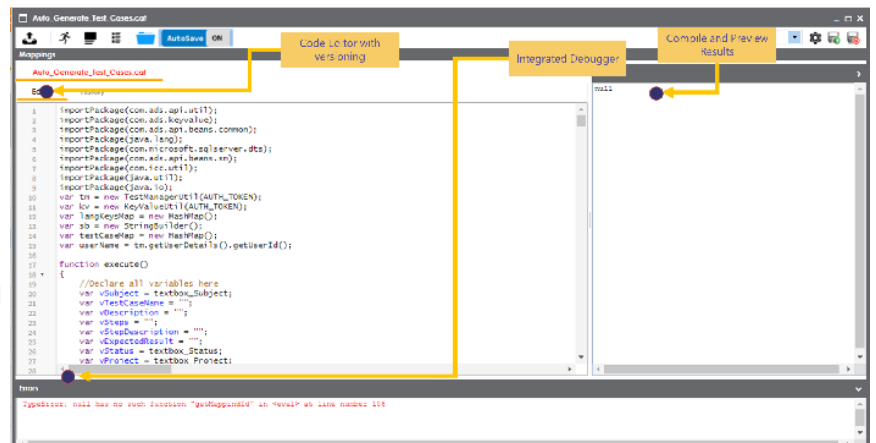
The erwin Automation Framework compliments erwin Mapping Manager. Our code automation templates (CATs) speed up manual processes and create high-quality, consistent code. Metadata-driven code engineering accelerates ETL/ELT development and cross-platform deployments and migrations, Big Data deployments, Data Vault, Data warehouse modernisation, data movement QA/testing tasks, SQL/DDL generation and more.

## KEY BENEFITS/RESULTS

- Centralised and standardised code management with all automation templates stored in a governed repository
- Better quality code and minimised rework
- Data movement job designs
- Flexible and future proof data preparation, deployment and governance
- Cross-platform support of scripting languages and data movement technologies

## KEY FEATURES

- Generation of ETL code based on standardised designs
- Generation of Big Data automation scripts
- Automated test-case generation
- DDL/SQL and procedural code generation
- Data Vault methodology templates and code generation
- Reverse-engineering of code into mapping documents
- Business-driven data movement and transformation specifications



## SOLUTION OVERVIEW

- Integrated development environment – SDK scripting languages and published libraries are used to automate tasks, such as generating DDL, ETL jobs and test SQL
- Centralised solution for ETL, SQL, DDL & Big Data Automation to ensure standards compliance
- Metadata-driven code generation of ETL/ELT jobs for Informatica, DataStage, Talend, SSIS, Pentaho, ODI, OWB, BODS and others
- Pre-built and customisable CATs to automate code generation for the Data Vault methodology, including hub, link, satellite code generation, table structures, ETL processes and DBMS views
- Code automations for SQL code and stored procedures, DDL from metadata to build warehouses and marts, JavaScript, JRuby, XSLT, Groovy and J-Python plus any Java-based external jars/libraries
- Standardise design patterns, including Stage-Load, SCD Dimensions Type 1, 2 & 3, fact patterns
- Support for test-case and test-SQL generation, validation- script generation, test-DDL generation, testfile generation and other associated scripting
- Big Data scripting support for PIG, Scoop, MapReduce and Python
- Discover, parse and interpret complex ETL, ELT and procedural code to reverse-engineer mapping documents and data lineage - accelerating the portability of data movement jobs across platforms and aids lineage creation automation

