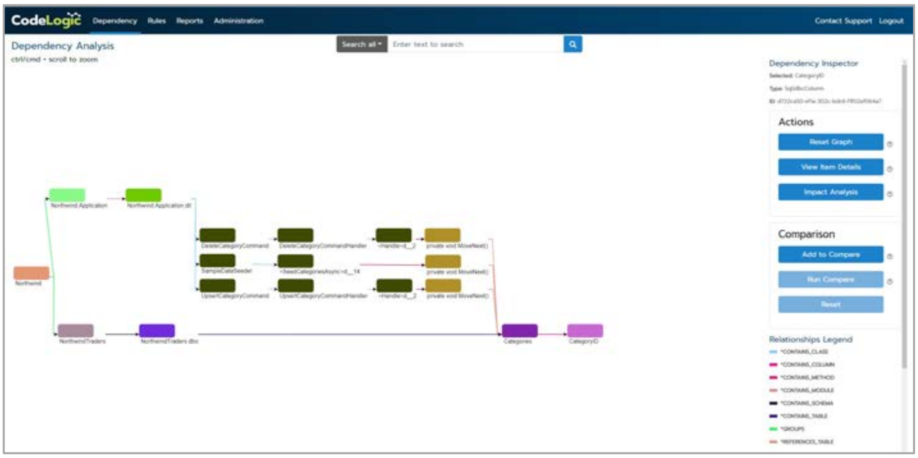
See Your Code Dependencies

More than ever, software engineers need a clear understanding of their application connections and dependencies to be more productive, better understand code complexity, and make informed application change and strategy decisions.



The Dependency Analysis view shows code dependencies from API to method to database column.

Make Code Changes With Confidence

CodeLogic provides the industry's most comprehensive, self-updating application code dependency mapping and analysis tool. CodeLogic goes beyond source code and starts where other tools — such as APM, ITSM, and IDEs — stop.

Teams using CodeLogic can Identify, visualize, analyze and understand application code complexity from API to method to database column. As a result, developers can reduce break/fix cycles, inform task completion timelines, improve team collaboration, and deploy code changes with confidence.

Reduce Break/Fix Cycles

Static & dynamic scanning

One-click impact assessment

IDE integration

- Speed up code reviews
- Improve task estimation
- Onboard team members
- Expose dependency risk
- Understand code impact

How Teams Use CodeLogic

See the Impact of Code or Schema Changes Before They're Deployed

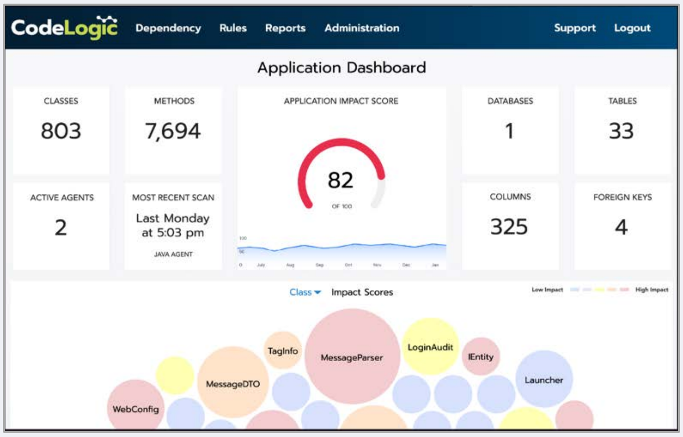
Tools, such as your IDE, only expose project-specific code dependencies. Our approach is different; CodeLogic exposes hidden code dependencies within and between applications and databases. We combine static binary scans with dynamic scanning to create a real-time, searchable system of record for code and database dependencies. This intelligence helps application teams see the impact of code and schema changes before deploying to production.

Make Better Rewrite or Refactor Decisions

Do you have any initiative to modernize legacy apps? Thinking about microservices? Wondering if the codebase is salvageable? CodeLogic makes evaluating an application's overall architecture and dependency complexity easy. Enabling teams to make informed rewrite/refactor decisions and quickly start on app modernization plans.

Identify the Technical Debt Worth Fixing

CodeLogic analyzes an application's code dependencies to find where code complexity and heavy usage intersect. CodeLogic identifies common forms of tech debt, such as orphaned methods and database columns, which add complexity without adding value.



Application Dashboard

CLASSES 803

METHODS 7,694

APPLICATION IMPACT SCORE 82 OF 100

DATABASES 1

TABLES 33

ACTIVE AGENTS 2

MOST RECENT SCAN Last Monday at 5:03 pm JAVA AGENT

COLUMNS 325

FOREIGN KEYS 4

Impact Scores

The Application Dashboard highlights important metrics.

Key Features

Static Dependency Scanning

Scans application binaries to discover their structure and identify dependencies between classes and methods. In addition, static dependency scanning finds database tables and columns and their relationships throughout an application.

Dynamic Dependency Scanning

Profiles an application as it runs in a test environment. This scanning method detects calls & identifies relationships between services or applications that cannot be found statically.

Application Dashboard

Summarizes key application metrics such as the number of code classes, methods, database tables, and database columns. In addition, the Application Dashboard highlights hotspots in an application's architecture by analyzing and detailing the top classes and methods with a high volume of dependencies.

Key Features

1-Click Impact Assessment

Quickly understand the full impact of a code or database schema change before deploying the change to production. CodeLogic 1-click impact assessment instantly illustrates how a proposed change impacts dependent API, classes, methods, and database columns.

Impact Scoring

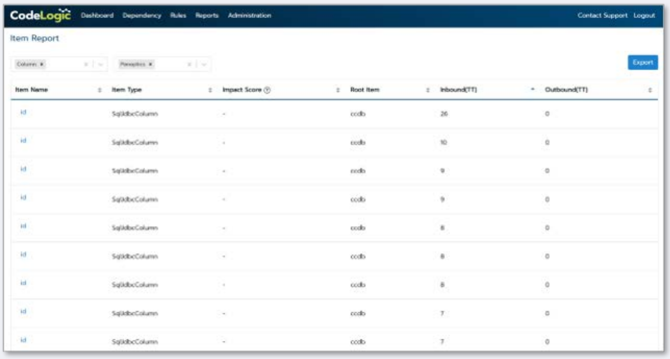
Reveal the health of problem areas in application code. The score is calculated by analyzing classes and methods to detect highly coupled code. The impact score also factors in code traits that indicate issues such as long methods and high complexity.

Application-to-Application Dependency Mapping

Captures relationships between applications, whether they are REST service calls or shared databases. Scanning more applications with CodeLogic makes it easier to see how applications depend on each other more clearly.

Dependency Visualization

Enable developers to explore application code dependencies in a graphical format.



Item Report

Item Name	Item Type	Impact Score	Root Item	Inbound(TT)	Outbound(TT)
id	SqlJdbcColumn	-	ccdb	26	0
id	SqlJdbcColumn	-	ccdb	10	0
id	SqlJdbcColumn	-	ccdb	9	0
id	SqlJdbcColumn	-	ccdb	9	0
id	SqlJdbcColumn	-	ccdb	8	0
id	SqlJdbcColumn	-	ccdb	8	0
id	SqlJdbcColumn	-	ccdb	8	0
id	SqlJdbcColumn	-	ccdb	7	0
id	SqlJdbcColumn	-	ccdb	7	0

CodeLogic reports make it easy to identify which database tables & columns have the most connections to an application.

Dependency Reports

Make it easy to create and export a list of relationships between classes, methods, and database columns. For example, teams can generate a list of every method or field that references a refactored database, or a list of every API endpoint exposed by an app moving to a new network. Dependency reports can be exported to a CSV and easily shared.

IDE Plugins and Extensions

CodeLogic integrates with popular IDE tools such IntelliJ and Visual Studio. Our Plugins and Extensions augments native usage and references provided within these tools and exposes references that are otherwise missed across projects and applications.

Open REST API

CodeLogic is built on an Open REST API facilitating the use of CodeLogic sourced dependency intelligence in other authenticated systems. The REST APIs are well documented in Swagger.

CodeLogic

Problems We Solve

Improve Cloud Migration Projects

Cloud migration can be a big undertaking for developers. Refactoring code and moving large amounts of data to the cloud can be difficult to plan. Not having a clear understanding of the connections within and across applications and databases further complicates the process — costing businesses time, money, and valuable resources. CodeLogic provides key insights into the connections and relationships across the applications and databases that are being moved, ensuring that your cloud migration process is a smooth success.

Improve Database Migration Projects

To successfully perform a database platform migration, teams must be able to see all the connection points between the app and database. Without an accurate, up to date view of all the app to database connections and dependencies, database migration efforts are likely to be costly, time consuming and prone to failure. CodeLogic visualizations and dashboards make it easy see every database connection across the application landscape. This enables project team members to better estimate the platform migration effort and develop a realistic migration plan.

Improve Developer Documentation

CodeLogic's ability to automate and deliver application dependency intelligences improves developer productivity by keeping track of application dependencies, illustrating the impact of code changes across the application landscape. That means developer documentation is always current and team members can easily understand how all parts of their app fit together. Now developers can spend less time crawling through the code and more time coding.

How CodeLogic Works

1. CodeLogic installs an agent on your servers where it extracts code, application and database metadata.

2. The agent sends metadata to the on-premise CodeLogic server, analyzes the data and generates a complete application map.

3. CodeLogic's web and desktop interface provide access to interactive application visualizations, analysis tools, and insights.

Learn More

www.codelogic.com