

Enhancing Software Code Quality for Common Data Service (CDS) Automation using BYOT and Parallel Execution

Raja Mohammed Hussain Peer Mohammed
Wipro Limited
Bellevue, WA, USA

Abstract — This white paper explores the integration of automation within Microsoft's D365 Common Data Service (CDS) a.k.a Microsoft Dataverse for Apps, focusing on enhancing software quality through improved testing methodologies. It highlights the benefits of centralized data management, streamlined app integration, and the ability to build workflows without coding. Key challenges addressed include high test flakiness, lengthy test execution times, and a lack of test analytics. Solutions such as Bring-Your-Own-Tests (BYOT) migration, telemetry logging, and code coverage automation significantly improved test stability, reducing test execution time from 15+ hours to 3 hours. By removing redundant tests and enhancing telemetry with Power BI reporting, organizations experienced better test performance and infrastructure stability. This paper outlines how these advancements contribute to higher software quality, faster testing cycles, and improved business outcomes, while offering insights into future enhancements for targeted test execution based on code changes

Keywords — Azure DevOps (ADO), D365, Common Data Service (CDS), Microsoft Dataverse, Bring-Your-Own-Tests (BYOT), Test Execution Optimization, Code Coverage, Telemetry

I. INTRODUCTION

As businesses scale and embrace digital transformation, the need for robust, scalable, and efficient software quality assurance practices is paramount. Microsoft's D365 Common Data Service (CDS) for Apps is an Azure-based platform that simplifies building, managing, and extending business applications by centralizing data from various systems like D365, SharePoint, and Office 365. This white paper demonstrates how leveraging automation within Microsoft's D365 Common Data Service (CDS) for Apps can significantly enhance software quality, reducing test execution times from over 15 hours to just 3 hours. With the integration of BYOT (Bring-Your-Own-Tests) and real-time telemetry, businesses can ensure faster, more stable releases, giving them a competitive edge in an increasingly digital world and enhancing test stability, optimizing test execution, and providing deeper analytics.

II. AUTHENTICATION METHODS IN AZURE DEVOPS

CDS for Apps is designed to centralize business data and facilitate app integration. By leveraging the Common Data Model (CDM), organizations can eliminate the need for extensive custom integrations.

Dynamics 365 applications—such as Dynamics 365 Sales, Dynamics 365 Customer Service, or Dynamics 365 Talent—also use Dataverse to store and secure the data they use. This

enables you to build apps by using Power Apps and Dataverse directly against your core business data, which is already used within Dynamics 365, without the need for integration

- Build apps against your Dynamics 365 data – Build apps quickly against your business data within Power Apps or by using the extensible platform that lets pro developers programmatically interact with data and metadata, apply business logic, create custom connectors, and integrate with external data.
- Manage reusable business logic and rules – Business rules and logic already defined in your Dataverse tables are applied to apps created with Power Apps to ensure data consistency, regardless of how your users access the data or which app they use.
- Reusable skills across Dynamics 365 and Power Apps – Users who are skilled in Power Apps or Dynamics 365 can now take advantage of those skills across the Dataverse platform. Creating tables, forms, and charts are now common tasks you perform across your applications

A. The Big Picture

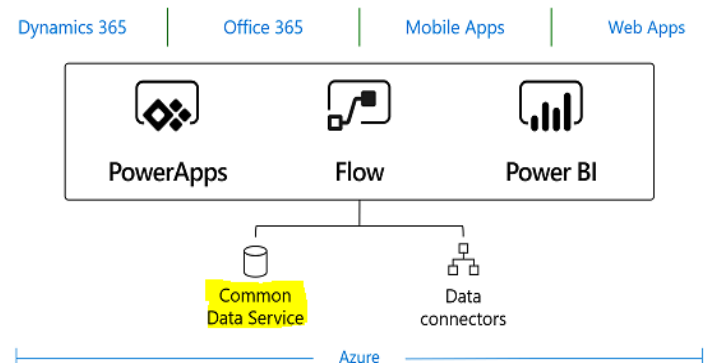


Fig. 1. Big picture – Overview of CDS Ecosystem

B. Common Data Model

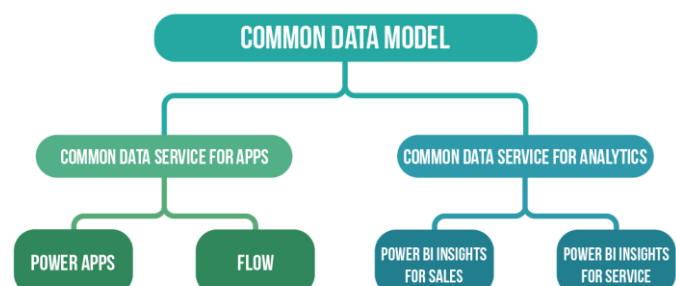


Fig. 2. Coomon Data Model – Building Blocks

III. KEY BENEFITS OF CDS

- A. Centralized Data Management
Both Metadata and Data is stored centrally in a cloud based server.
- B. High Data Security
Role-based security allows to control access to tables for different users within the organization. Users can see the data only if access is granted to them
- C. Application Integration
Seamless integration with Microsoft apps and third-party services (e.g., Salesforce, Google Drive).
- D. App Creation and Workflows
Users can build apps without coding, enhancing agility in business processes.
- E. Logic and validation
Define calculated columns, business rules, workflows, and business process flows to ensure data quality and drive business processes
- F. Advanced Analytics
Collecting useful insights from business data.

IV. ADDRESSING CHALLENGES IN CDS TEST AUTOMATION

As organizations increasingly rely on software-driven processes, maintaining high test accuracy and efficiency becomes crucial. The CDS automation approach focuses on tackling key pain points in testing:

- A. Test Flakiness
High levels of flakiness in test results reduce reliability. Current pass rate is less than 70%, which is way below the industry standards.
- B. Integrating 3rd Party apps
3rd Party app developers must wait till the Beta is released to test the compatibility of their apps with newer version of Dynamics 365. This significantly reduces the time to market of newer versions.
- C. Parallel execution
This allows organizations to simultaneously test across multiple environments, reducing the overall cycle time and accelerating the release of updates. Current time taken for completing one cycle is more than 15 hours
- D. Telemetry Logging
This enables the correlation of test data, facilitating detailed insights into performance and stability, helping to refine test cases further

V. KEY AUTOMATION SOLUTIONS IN CDS TESTING

- A. BYOT Migration
CDS introduced Bring-Your-Own-Tests (BYOT) to allow users to run their own tests within Microsoft's pre-release infrastructure for validations like PR/BVT/Acceptance testing. Key benefits include:

- Test execution against various online infrastructures such as TIE and TIP.
 - Transition of test scripts from executable files (EXEs) to dynamic link libraries (DLLs) enabled more flexible, parallel testing.
 - The BYOT setup removed 18.5K redundant tests, reducing the codebase footprint by 50%.
- B. Test Telemetry and Analytics
With telemetry logging enabled across 1,250+ tests, CDS enhanced monitoring by implementing unique correlation IDs for each test case. This telemetry integration allows a tree-view report in Power BI, enabling better visualization and tracking of test execution metrics. Additionally, real-time error reporting through Kusto DB and event-based triggers allow for more agile responses to test failures.
 - C. Testcase Telemetry & Reporting
Test telemetry was extended to include activity vector incrementing within each test case, further refining analytics. Coupled with Power BI integration, stakeholders can now access detailed historical and real-time reports.
 - D. Code Coverage (CC)
Automation of platform signoff tests included the execution of 3,325 tests for code coverage, of which 3,317 passed. Test coverage files were merged for better reporting, providing a foundation for comprehensive code analysis. This proactive coverage enables businesses to monitor any changes in code quality dynamically.

VI. BENEFITS OF CDS PLATFORM TEST AUTOMATION

- A. Reduction in Redundancy
The removal of unnecessary tests not only reduced the testing footprint but also contributed to improved test execution times.
- B. Improved Stability
The CDS repository infrastructure's stability improved from 70% to 95%, directly contributing to more reliable software releases.
- C. Faster Testing Cycles
Parallel execution across multiple environments allowed for quicker testing, accelerating the time-to-market for software updates.

VII. CASE STUDY – EFFICIENT TEST EXECUTION IN AZURE DEVOPS USING BYOT AND PARALLEL EXECUTION

- A. Pain Points in the project before implementation
 - High Test Flakiness and need to improve pass % to >99.8
 - Unique correlation of product telemetry not available
 - Migrate all Test execution to BYOT and run them against TIE/TIP Infra (Online Orgs)
 - Code Coverage info is not available.
 - CDS Repo Infra health was unstable and impacting Dev productivity

B. Solution Implemented

- Improved the pass percentage to 99.8% (frequently hitting 100% pass rate) by fixing 12 Test Patterns (covering almost 50+ tests)
- Enabled Telemetry logging on all the 1250+ tests that are part of Platform signoff suite
- Implemented unique correlation ID for each TC to identify the Product Telemetry
- Implemented BYOT (Bring-Your-Own-Tests), which enables users to write their own tests and run them in MSFT provided infra, for pre-release testing, like PR/BVT/Acceptance validations
- Convert all the tests to DLLs instead of the current EXEs. This enables us to run tests in BYOT
- Automated the platform signoff tests on OnPrem environment for Code coverage.
- 3325 tests were enabled for CC and executed.
- Generating Automatic alerts, Auto-mitigation using Microsoft Flow & Telemetry

C. Business Benefits and Return-on-Investment

- Removed around 18.5K+ tests that were not being run at all. This helped in reducing the code footprint by more than 50 %
- Telemetry logging enables a Tree-View for Power-BI reports
- Enabled Parallel test execution using multiple orgs, thereby reducing the total execution time from the current 15+ hrs. to about 3+ hrs. now
- Repo Infra health improved from ~70% to ~95% stability

D. Migration to Bring-Your-Own-Test (BYOT) concept

- BYOT (Bring-Your-Own-Tests) enables users to write their own tests and run them in MSFT provided infra, for pre-release testing, like PR/BVT/Acceptance validations
- Convert all the tests to DLLs instead of the current EXEs. This enables us to run tests in BYOT
- Removed around 18.5K+ tests that were not being run at all. This helped in reducing the code footprint by > 50 %
- Tests are now run against online flavors of CRM like TIE\TIP infra
- Complete error reporting for each test case, including Event Logs with crash data, are now available in Kusto DB
- Event-based Automated trigger of Test runs like Product updates, DB Updates, Solution installs etc.

E. Code Coverage

- Automated the platform signoff tests on OnPrem environment for Code coverage.
- 3325 tests were enabled for CC and executed; 3317 passed; only 8 failed
- Created dedicated Org and different users to execute the multi org tests and tests that require specific user roles like salesperson
- As and when new TCs are added to the repository, CC is enabled for them as well.
- Coverage files for most relevant dlls have been obtained and merged into a single coverage file

F. Testcase Telemetry and Power BI reporting

- Enabled Telemetry logging on all the 1250+ tests that are part of Platform signoff suite
- Implemented telemetry logging with single correlation id and increment the ActivityVector within each TC – this enables a Tree-View for Power-BI reports
- Implemented sample Power-BI report with metrics requested by customer

G. Sample Power BI Reports

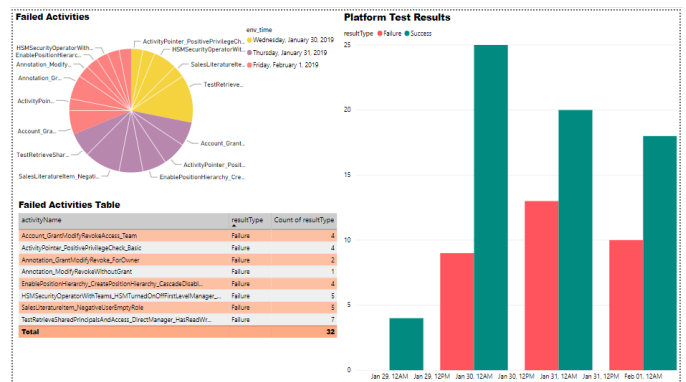


Fig. 3. Sample Power BI report – Test Results

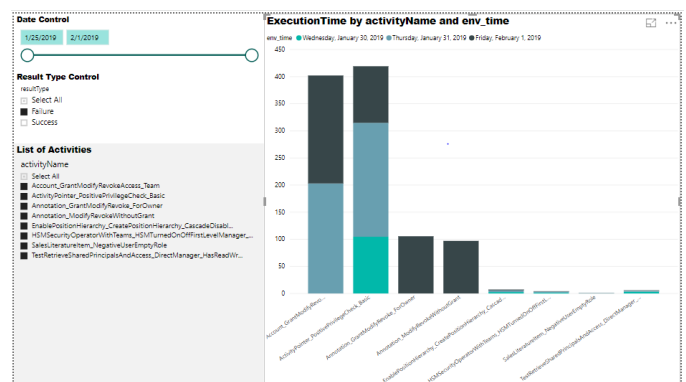


Fig. 4. Sample Power BI report – Execution time

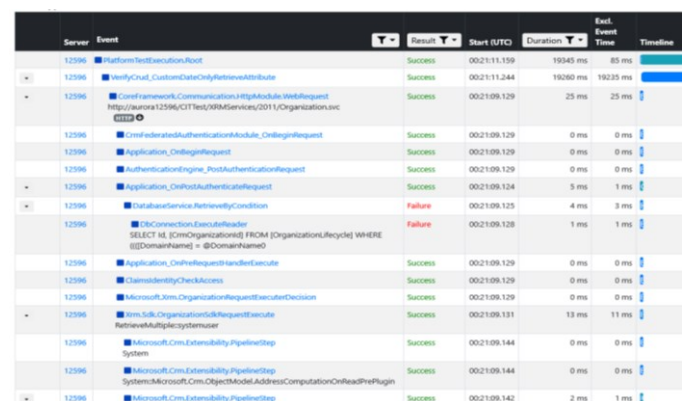


Fig. 5. Sample Power BI report – Tree view by CorrelationID

H. Measurements and Metrics

Pain Point	Before Implementation	After Implementation	Benefits
High Test Flakiness	Pass % less than 70%	Pass % greater than 99.8% (frequently hitting 100%)	Improved Test Reliability
Code Coverage	Data Not available	Achieved >90% coverage which is higher than the org norm	Coverage Data and improved code coverage
Repo Infra health was unstable	Unused Test code	Identified and removed 18.5K+ redundant tests	Improved Dev productivity and reduced code by >50%. Repo Infra health improved from <70% to >95% stability
Test Execution is taking too long	>15 hrs.	Enabled parallel test execution and total time taken is <3 hrs.	>80% savings in time and resources. Enabling faster releases
Test Analytics data	Not available	Enabled Telemetry logging on all the 1250+ tests. Implemented unique correlation ID for each TC to identify the Product Telemetry	ActivityVest or enables a Tree-View for Power-BI reports

Table 01 - Sample Power BI report – Execution time

VIII. FUTURE ROADMAP

To further enhance software quality, CDS automation continues to evolve:

- Customized Test Runs Based on Code Changes: Future test execution will map test cases to specific code changes, allowing for even more targeted and efficient test executions.
- DLL-to-Test Mapping: A roadmap to enhance test management by creating direct links between DLLs and test cases, ensuring tests are streamlined and relevant.
- AI-powered test automation will enable organizations to dynamically allocate resources based on predictive insights from historical test data, further improving efficiency and reducing errors

IX. CONCLUSION

Automation in the D365 Common Data Service (CDS) for Apps has delivered significant improvements in software quality engineering, offering a scalable, efficient, and highly reliable solution. The combined advantages of BYOT migration, telemetry logging, and improved code coverage have streamlined test execution and enhanced overall platform stability. By leveraging CDS automation, businesses can streamline testing, enhance stability, and reduce operational costs, resulting in faster product delivery and higher customer satisfaction. These improvements not only optimize internal processes but position organizations for long-term success in an increasingly competitive digital marketplace.

REFERENCES

- [1] Microsoft Docs, "Common Data Model." Microsoft, 2023.
- [2] Microsoft Docs, "Microsoft Dataverse." Microsoft, 2023.