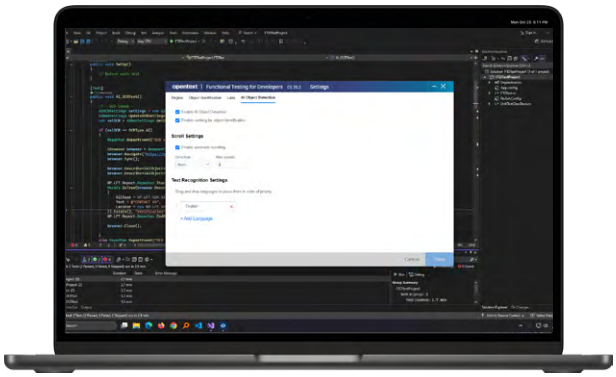
Top 10 reasons OpenText Functional Testing for Developers is a must-have



Constantly changing applications, diverse technologies, and the need for swift, high-quality results when developing software can be daunting.

Unlock the power of seamless software delivery and discover how OpenText™ Functional Testing for Developers tackles these pain points head-on. Dive into the top ten reasons why this shift-left functional test automation tool is essential for DevTesters and engineers, accelerating productivity and ensuring code quality in every development cycle.

1. **Shift-left testing:** Accelerate development by testing simultaneously with application creation, enhancing collaboration and boosting productivity.
2. **Multi-language support:** Write test scripts in Java, JavaScript, or C#, ensuring flexibility and ease for developers using diverse languages.
3. **IDE integration:** Seamlessly integrate with popular IDEs like Visual Studio Code, Visual Studio, Eclipse, and IntelliJ so developers can work in their preferred environment.
4. **AI-based testing:** Using OpenText™ Aviator™ AI capabilities, identify UI objects with combo box support, pre-operation object verification, and exact match or regex-based text recognition, boosting test resilience and reducing maintenance overhead.
5. **Enhanced Selenium integration:** Expand Selenium's power with OpenText Functional Testing for Developers API extension, facilitating quick and robust test script creation.
6. **Comprehensive reporting:** HTML run reports provide detailed insights into test execution, aiding developers in swiftly identifying errors and improving code quality.
7. **CI server compatibility:** Support for Jenkins, TeamCity, Bamboo, and Azure DevOps ensures effortless integration into continuous integration pipelines. With JUnit, format result reporting and user-account controls for streamlined, flexible workflows.

Highlights

- Boost productivity and accelerate delivery with shift-left test automation tools.
- Speed up test execution while improving code quality.
- Improve visibility of automated test execution with quality reports.
- Increase developer efficiency with a broad ecosystem of integrations.
- Develop and run tests from Visual Studio Code across Windows, macOS, and Linux.
- Accelerate test resilience with AI-based object identification and smarter text recognition.

8 **Object identification tools:** Maintain consistency in testing models by adapting to changes in the underlying application with full UI hierarchy navigation, relative object paths, and access to OpenText™ Functional Testing methods and properties across all SDKs.

9 **Parallel testing:** Run multiple tests concurrently on various environments, physical, virtual, or dockerized, to ensure efficient use of time and resources.

10 **Mobile testing integration:** Seamlessly integrate with OpenText™ Functional Testing Lab for Mobile and Web for end-to-end automation of mobile applications, providing developers with comprehensive mobile testing capabilities.

Learn more about OpenText Functional Testing for Developers ›

For information on how to get started and what's new, visit the help center ›

> **"Leveraging OpenText solutions, we have increased our test coverage as we can test with more users, and against different datasets, resulting in a higher quality application for our customers.""**
>
> **Test Automation Project Lead**
> Large Aeronautic Manufacturer

Copyright © 2026 Open Text • 08.26 | 269-000044-001