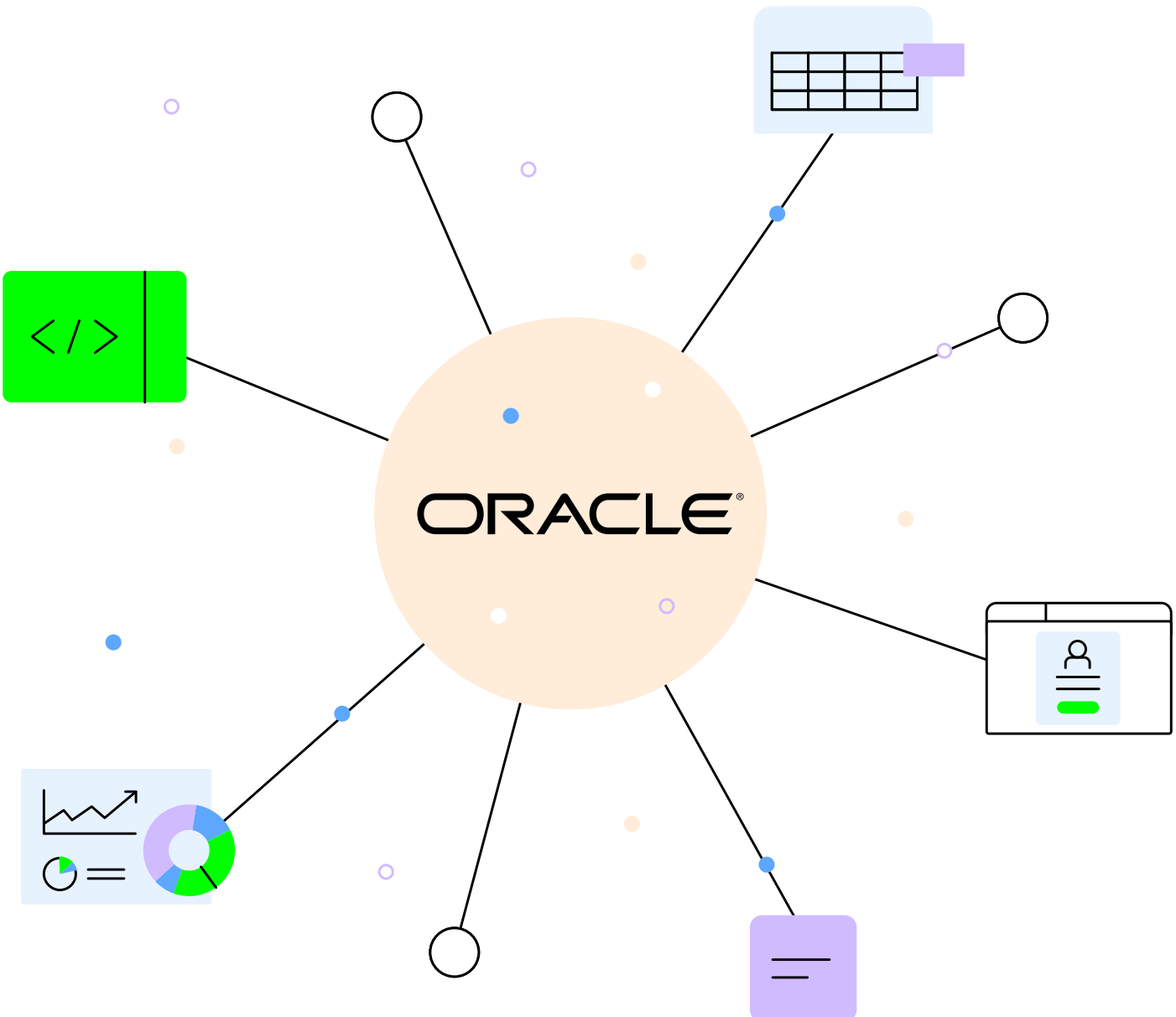
leapwork

Automating Oracle

Everything you need to know about test automation for Oracle



Oracle is one of the largest enterprise software vendors in the world, providing global businesses with databases, ERPs and other solutions for core business processes to run on.

Ensuring that these processes run efficiently and error-free is of the utmost importance, and has only become more critical as digitalization becomes a necessity for most businesses.

Automation can help mitigate risk, increase efficiency and reduce costs by helping QA teams verify at speed that applications and business processes run as intended.

In the following, we'll take a closer look at how test automation can be leveraged by QA teams to ensure quality delivery at speed.

Table of contents

Oracle Test Automation

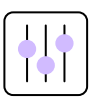
For businesses using Oracle software, testing is a crucial element in ensuring that business-critical processes run as intended in and between Oracle applications and the business' remaining ecosystem.

When performed manually, testing is not only extremely time-consuming, but also tedious and error-prone. This is because tests should be executed every time a new process is set up, every time a process is adjusted, and every time Oracle updates their system. That adds up to a lot of testing. Test automation is the only way for QA teams to keep up with testing requirements without compromising quality.

Applying Test Automation in Oracle is particularly critical for the following reasons:

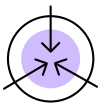
Frequent updates

Businesses must be able to frequently change and update process flows to reflect changes in business strategy and to keep up with market demands. The more changes, the higher the risk of breakage, making frequent testing crucial.



Customizations

Oracle is highly customizable to match the business' individual needs and requirements. The higher the level of customization, the higher the risk of breakage when updates occur or changes are made, again making testing crucial.



Integrations

Oracle rarely stands alone in a business' ecosystem, meaning integrations are typically set up between Oracle applications and other applications. This increases the complexity of the system, making the argument for testing stronger.

Growing data pool

While the pool of transactional data across systems grows, so does the need to test and ensure that data is flowing and stored as intended.

By automating tests, QA teams can increase test coverage and complete testing within a shorter time frame. This can give businesses the confidence that their most critical processes work, and that they will be notified if something doesn't, so that they can fix it before it affects the end-user or has operational or financial consequences or becomes a security or compliance threat.

The problem is that setting up and maintaining automation is, to many QA teams, just as time-consuming (at least in the short term) as testing manually. In this way, test automation can seem like a bit of a paradox - it can save you time once you have it, but getting to the point of time-saving is extremely time-consuming.

This is why finding the right tool is essential to success with test automation.

What to look for in an Oracle test automation tool

For most enterprises, Oracle is a part of a larger ecosystem of technologies and platforms that, together, make up the business' core processes.

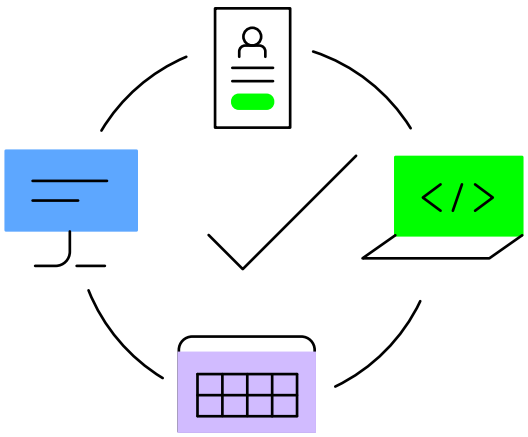
This has implications for the automation tool; it must be robust and reliable, it must be capable of integrating multiple technologies, and it shouldn't add additional complexity to an already complex web of systems. This adds up to a number of features that your Oracle automation tool should have. These include:

1.
Cross-technology functionality,
allowing businesses to test across all integrations

2.
Intuitive and visual language rather than code,
making automation design and maintenance easy

3.
Robust GUI controls recognition and Oracle Forms automation capabilities,
solving common maintenance burdens and technical challenges

In the following, we'll have a closer look at each of these features, and shed some light on why these are important for your future Oracle automation platform.

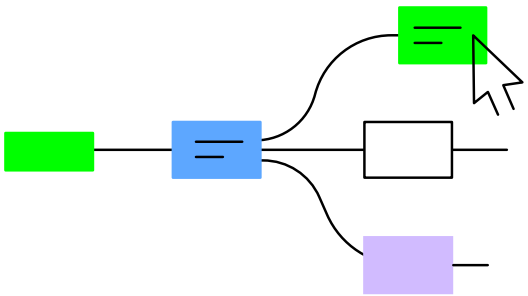


1. Cross-technology functionality

The first feature, cross-technology functionality, is critical because, although Oracle is central to your business, your business processes probably span across multiple applications, even within a single business transaction. If your tool doesn't allow you to automate across technologies, you may have to use several partial automation solutions under one roof, which only adds additional complexity to your system architecture.

What's more, because Oracle in and of itself consists of multiple applications, spanning from legacy systems and desktop to newly developed applications and web, the chosen tool should be able to work across all these. This is the primary reason why Selenium, the popular open-source web automation tool, isn't a good candidate for Oracle automation.

It goes without saying that if the goal is to create seamless and efficient end-to-end tests, that will let you validate not just single applications, but full processes, and give you full confidence in the quality of these, then cross-technology functionality is a must-have.



2. Intuitive and visual language rather than code

The second feature, no-code, is important for keeping testing fast and agile. Automation setup can be both time-consuming and challenging if it requires coding. Maintenance can be a major burden if it requires re-coding scripts for automation.

If you do choose to create automation as script, there's a risk that your testing will eventually become extremely time-consuming and challenging to maintain for non-coders as your testing suite grows, and eat away at your available tester and developer resources.

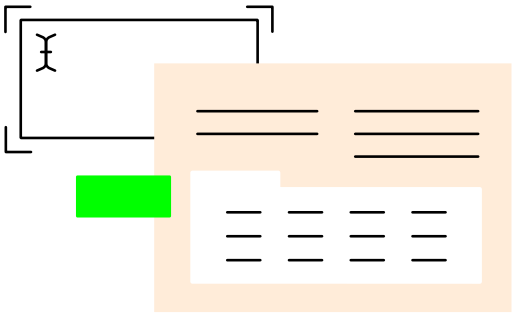
No-code can solve this problem by giving you not only a shorter learning curve and faster setup, but also a clear overview of your testing suite once it is set up, that will let you make changes and updates quickly and easily.

With a good no-code automation tool, anyone in the organization, whether they are business experts or technology experts, can understand and contribute to automation.

Business users can automate

Easy maintenance

Scalable



3. Robust GUI controls recognition and easy Oracle Forms automation

As mentioned above, maintenance is one of the biggest burdens in test automation, and particularly in Oracle systems due to their size and complexity, and so finding ways to lessen this burden can save teams from a lot of tedious work.

A robust GUI control recognition is a technical capability that ensures that the automation tool is able to find elements, even when they change slightly, which means less maintenance in the long run.

Another common challenge in Oracle automation is Oracle Forms. Oracle forms is Oracle's own technology to design and build enterprise applications. It often runs in the background of applications and serves as an enabler of important actions. For example, it could be the background technology that lets a user submit a query or add customer information to a Bank's Customer Services System.

Oracle forms are developed on Java applet technology and sit as just one object inside the DOM, making it difficult to identify the form's individual elements within that web object. For this, you need a tool that can go beyond the web element (which Selenium won't allow you to), and read what's inside the web object, allowing you to browse, interact and perform operations with Oracle Forms.

Another reason Selenium isn't optimal for automating web applications built with Oracle Forms, is that Oracle Forms are only supported by IE - even though IE has been discontinued - and it is notoriously difficult to automate IE with Selenium. The IE driver in Selenium isn't built by Microsoft and won't provide you with the stability and depth of testing that anyone using Oracle forms would need.

Tools for Oracle Test Automation: Selenium, OATS, and Leapwork comparison

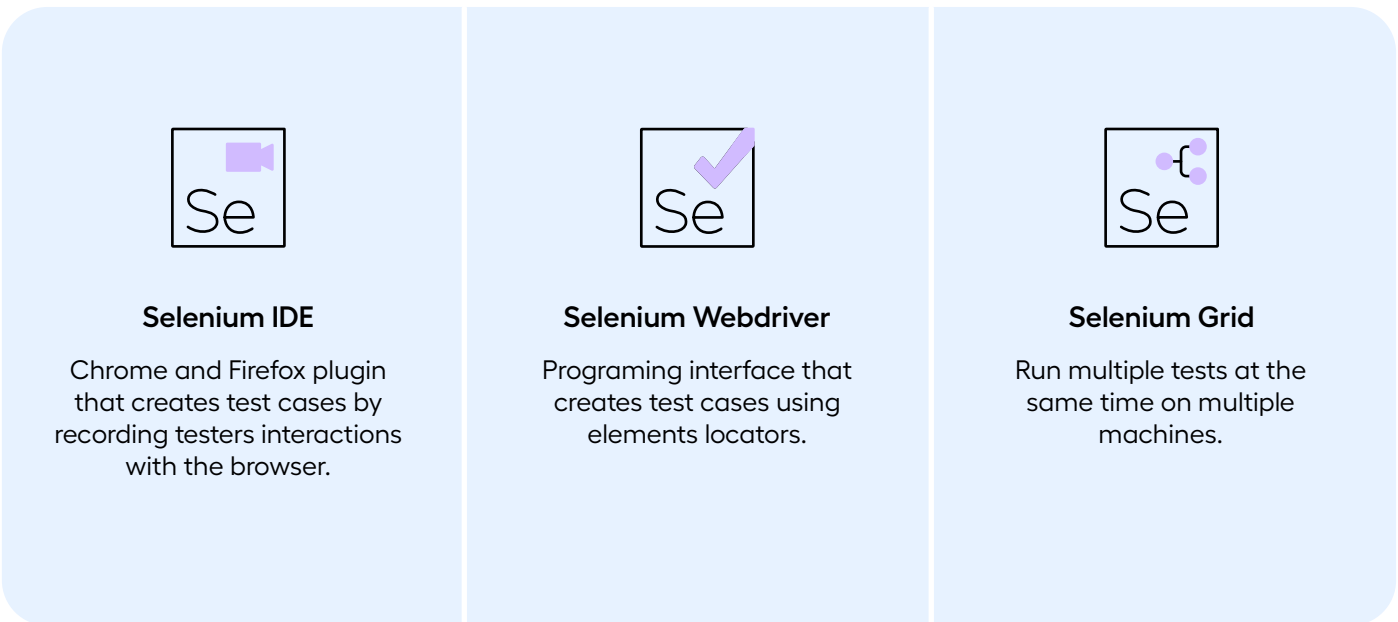
When researching options for an Oracle test automation tool, there are a number of things to consider, such as what capabilities the tool should have and if the tool should be open-source or licensed.

In the following, we'll give an overview of three Oracle test automation tools and discuss the pros and cons of these, in order to give you a better understanding of your options.

Selenium

Selenium is a highly popular test automation tool for several reasons; it's free and open-source, it has a large community of users, and it can be used for web GUI testing.

Selenium tool suite



The Selenium automation suite consists of several tools, including Selenium IDE which is a record and play tool that makes test setup easier, and Selenium Grid, which lets you perform parallel testing.

The downside of Selenium is that you need to code to set up and maintain your tests, which makes testing with Selenium a tedious and time-consuming affair in the long run - particularly if your system is like many other enterprises' using Oracle: large and complex. It also only works with browser-based applications, meaning you won't be able to automate across any desktop-based Oracle applications or other non-web technologies.

On top of being difficult to maintain, Selenium also doesn't provide you with any test reporting or troubleshooting capabilities, making failed tests difficult and timely to fix.

Last, because it's open-source, you won't be able to get fast and targeted professional help if you run into technical problems, which can be a deal-breaker for enterprises dealing with sensitive operations.

Oracle Application Testing Suite (OATS)

Oracle has its own test automation tool called Oracle Application Testing Suite, but often referred to as OATS, which is to Oracle users an obvious test automation tool candidate.

It probably comes as no surprise that OATS is fully compatible with Oracle's own applications. It also comes with built-in automation components that can be used with Oracle's applications, which eases the test design and setup process.

OATS does, however, also have its limitations. The first is that, like Selenium, it is limited in it's cross-technology functionality, and it will only allow you to automate Oracle and web-based products.

Unlike Selenium, and other open-source tools, however, it does offer some premium features, such as scheduling for regression testing and test data creation.

Still, the limitations to cross-technology capability mean that if you choose OATS, you will most likely need additional automation tools to achieve complete test coverage and to have full end-to-end testing, which is the goal for most QA teams in enterprise organizations.

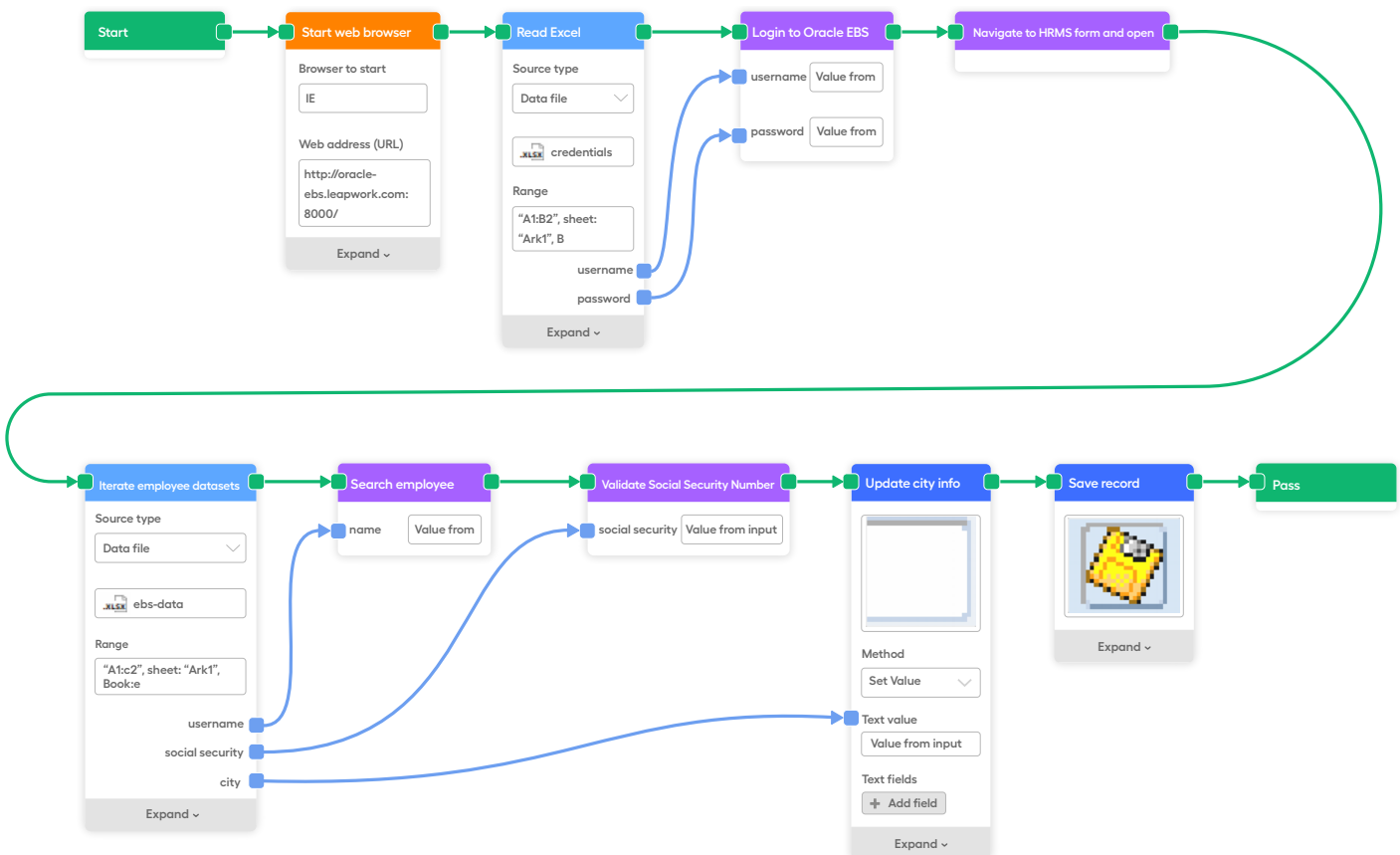
Leapwork

Leapwork solves the above problems by fulfilling the most critical needs for Oracle test automation, as well as offering additional features that are crucial to teams following an agile, continuous delivery approach.

With Leapwork's no-code automation platform, enterprises can run test automation across all Oracle-based applications. Leapwork also solves the common challenge of automating Oracle Forms by having built-in IE capabilities. In addition to these capabilities, Leapwork is intuitive and easy to use. Conventional Oracle automation requires a sophisticated understanding of the Oracle technology stack a s well as programming languages. This is a problem that Leapwork solves.

Leapwork's visual approach and flow-chart user interface removes noise and complexity that is common in other test automation platforms, making it is easier and faster for QA teams to become productive with automation. Users can tell Leapwork what to automate, without needing to express how to do it. And a knowledge of coding terminology or code isn't required to be successful.

Example of Leapwork automation flow for Oracle



With Leapwork, teams can use the same visual approach to automate and maintain tests across different applications, so organizations can reduce risk and guarantee business continuously in complex enterprise landscapes.

What's more, Leapwork has robust and reliable GUI controls recognition, it has building blocks for driving any type of flow with any type of data, and it has added advantages such as parallel execution support, cross-browser support, third-party web cloud deployment, dashboards and reporting.

To learn more about Leapwork and no-code test automation make sure to join our webinar by signing up below.

On-demand webinar

No-code Test Automation With Leapwork

Learn how to build your own codeless UI automation with the Leapwork Automation Platform.

Watch now

leapwork