

L'outil de test GUI pour Java, Web, Android & applications Windows natives

<div style="text-align: center;">  ✓ </div> <p style="text-align: center;">L'outil professionnel de l'automatisation de test GUI</p>		
Fonctionnalités		
Technologies GUI de l'application à tester	Applications Java : Swing, JavaFX. AWT, SWT, Eclipse plug-ins, RCP, Applets, JavaWebStart, RIA, ULC, CaptainCasa pour les composants JavaFX SubScene, JIDE Common Layer	
	Applications Web : Browser: Chrome, Firefox, Opera, Safari, Edge (Chromium basiert), Microsoft Edge Legacy, Internet Explorer; Versions Headless Browser pour Chrome, Firefox et Edge (basé sur Chromium) HTML 5, AJAX: QF-Test supporte complètement des frameworks comme Angular, React et Vue.js, mais aussi des toolkits concrets comme Smart (GWT), ExtGWT, ExtJS, ICEfaces, jQuery UI, jQueryEasyUI, Kendo UI, PrimeFaces, Qooxdoo, RAP, RichFaces, Vaadin et ZK. Autres toolkits peuvent être intégrés avec peu d'effort par exemple: SAP UI5, Siebel Open UI et Salesforce. Tester des applications Electron et Webswing est supporté aussi.	
	Applications Android : Applications Android natives, applications web mobiles et applications hybrides sur Android à partir d'Android 7 sur des appareils réels et avec l'émulateur d'Android Studio	
	Applications Windows : Win32 classique, .NET (souvent développé en C#), Windows Presentation Foundation (WPF), Windows Forms, Windows Apps / Universal Windows Platform (UWP) avec éléments de contrôle XAML, applications C++ modernes (p.ex. avec Qt)	
	Systèmes hybrides avec la combinaison des plusieurs technologies GUI, aussi des composants incorporés dans le navigateur (JavaFX WebView, JXBrowser, SWT-Browser)	
	Documents PDF peuvent être testés können wie eine normale Anwendung getestet werden (Verifications textuelles et graphiques pour des éléments individuels)	
Support GUI selon le système d'exploitation	Applications Java: Swing und JavaFX: Windows, Linux, Unix, macOS SWT: Windows, Linux-GTK; Solaris-GTK sur demande.	
	Applications Web: Windows, Linux, macOS	
	Applications Windows: Windows	
Principe de test	Capture/Relecture pour la génération directe et efficace de séquences de test pour un traitement ultérieur dans des cas de test plus complexes avec contrôle de flux, paramétrage, modularisation et capacités de script avancées. Tout peut être personnalisé.	
Structuration de test	Clair et concis grâce à la représentation graphique des cas de test et des nœuds d'action dans une structure arborescente. Les projets peuvent être structurés de manière modulaire en utilisant des suites de tests et des bibliothèques.	
Reconnaissance des composants, Robustesse des tests	Détection stable des composants, indépendamment des propriétés géométriques, même pour des éléments complexes comme les arbres et les tableaux dynamiques. Les tests sont robustes et tolérants aux modifications de l'interface graphique.	
Réutilisabilité, Effort de maintenance	Haute réutilisabilité des modules de test grâce à une structure modulaire, par exemple par des procédures, l'encapsulation des accès aux composants, etc.	

Tests basés sur des données	Importation directe de fichiers CSV ou Excel, utilisation de requêtes de base de données SQL, fichiers XML. Toute autre source peut être intégrée via une extension de script.	
Tests basés sur les mots-clés/ Tests basés sur le comportement	Utilisation de mots-clés (Keywords) pour mettre en œuvre et contrôler les cas de test, également au moyen de documents ou d'outils externes de spécification des tests (par exemple, Excel ou des outils de gestion des tests)	
Conteneurs Docker	QF-Test supporte les tests conteneurisés via Docker	
Tests de charge et de performance	Tests de charge et de performance grâce à une exécution synchronisée et parallèle, même sur plusieurs machines. Pour le web en combinaison avec des outils comme JMeter ou NeoLoad.	
Protocoles, Documentation des tests, Rapports	Des protocoles clairs et détaillés comprenant des captures d'écran de la situation d'erreur sont toujours générés. Des rapports configurables dans différents formats (HTML, XML, JUnit), la documentation des tests et des procédures peuvent être générés par simple pression d'un bouton ou automatiquement.	
JIRA /REST	JIRA et JIRA Plugins comme TestRail, Zephyr, X-Ray, TM4J peuvent être intégrés via REST, souvent aussi via des outils CI comme Jenkins.	
Extensibilité par des scripts	Extensions de fonctions libres et contrôles/actions personnalisés par des scripts intégrés (Jython, Groovy et JavaScript). Via l'API de script, accès complet à tous les objets de l'application (SUT) et exécution de son propre code dans l'application ou dans le navigateur.	
Gestion des tests	Fonctionnalités de base pour les petits projets inclus dans QF-Test.	
Intégration continue	Intégration disponible ou possible pour : ALM/QualityCenter de MicroFocus/HP, TestBench d'Imbus, QMetry, Klaros de Verit, TestLink, IBM Rational Quality Manager, Jira et Jira Plugins tels que TestRail, Zephyr, X-Ray, TM4J, entre autres.	
Bureaux virtuels	p.ex. Citrix, VMware, VirtualBox	
Gestion des versions	Bonne versionnabilité, par exemple via Git, SVN/Subversion, CVS, Mercurial, grâce au format XML des fichiers pertinents.	
Suivi des bogues	Possibilité de se connecter via des interfaces ouvertes et REST, p.ex. Jira, MantisBT, Bugzilla	
Robot Framework	Les mots-clés du Robot Framework peuvent être simplement mis en œuvre et exécutés en tant que procédures de QF-Test.	
Exécution des tests		
Préparation des tests	Un assistant de démarrage rapide permet de générer une séquence de démarrage adaptée à l'application testée, en fonction de la technologie GUI sous-jacente.	
Préparation des scénarios de test	Gestion des dépendances pour la préparation et le post-traitement des tests pour des cas de test exécutables indépendamment, y compris la gestion automatique des erreurs.	
Points de vérification	L'enregistrement direct des contrôles standard, les vérifications spécifiques au client peuvent être mises en œuvre de manière variable par du scripting.	
Comparaison d'images	Enregistrement direct des contrôles d'image possible. De nombreux algorithmes, y compris pour les comparaisons d'images floues, et une vue différentielle pratique pour la vérification en cas d'écarts.	
Cartographie des objets	Les informations sur les composants sont stockées de manière centralisée pendant l'enregistrement dans une zone dédiée de la suite de tests, modifiable à tout moment. Mécanismes de recherche de références et de mise à jour automatique.	

Localisation intelligente des objets	Avec les SmartIDs, les composants peuvent être adressés directement sur la base de propriétés caractéristiques, c'est-à-dire le label associé. La définition d'une portée permet de restreindre la recherche de l'objet, par exemple en cas d'éléments multiples.	
Composants génériques	Le mappage des composants spécifiques de l'interface graphique en composants génériques (boutons, champs de texte...) permet la réutilisation des tests entre les technologies ainsi que l'utilisation d'actions généralement valides sans avoir à capturer chaque composant individuel.	
Exécution du test via la ligne de commande	Exécution en mode batch possible avec des options de configuration étendues via des paramètres de ligne de commande, également pour l'intégration dans des environnements de construction.	
Exécution à distance	Exécution du test également sur des ordinateurs distants en mode daemon.	
Traitement des erreurs	Le traitement automatique des erreurs garantit la poursuite de l'ensemble du test sans interruption. Les erreurs sont enregistrées pour une analyse ultérieure.	
Débugueur de test	Fonctionnalité complète de débogage, y compris les points d'arrêt et l'analyse des variables.	
Convivialité		
Confort d'utilisation	Utilisation facile et intuitive avec une vue arborescente clairement structurée pour une édition pratique des cas de test, par exemple par copier/coller et glisser/déposer. Capture/relecture pour un démarrage rapide	
Connaissances préalables requises	Aucune connaissance en programmation n'est requise pour une utilisation standard.	
Travail en équipe	Pour les scripts, la connaissance des langages de script standard Jython, Groovy et JavaScript est utile.	
Licences et prix		
Variante de produits	QF-Test est disponible en différentes variantes de produits variables configurables pour les technologies d'interface graphique prises en charge (Swing, JavaFX, SWT, Web et Windows) - Achetez uniquement ce dont vous avez besoin.	
Types de licences	Licence développeur - pour créer (et exécuter) des cas de test. Et licence runtime - pour exécuter des tests (nocturnes).	
Mécanisme de licences	Toutes les licences sont flottantes (c'est-à-dire qu'elles ne sont pas liées à une personne spécifique). Les licences standard fonctionnent dans un réseau (local), pour une utilisation inter-réseau nous proposons un serveur de licences.	
Acheter ou louer	Nous proposons l'option d'achat et la location sur une base annuelle, pour les licences de test de charge également des termes plus courts.	
Support und Service		
Télécharger et tester gratuitement	Téléchargement gratuit et anonyme de la version démo. La version de démonstration peut être exécutée sans enregistrement de la	

Installation	Installation facile sous Windows, macOS et Linux en quelques clics.	
Support	Assistance directe des développeurs et des testeurs de QF-Test en français, anglais et allemand.	
Contrat de maintenance	Le contrat de maintenance (support + mises à jour) sur une base annuelle comprend du support par e-mail et par téléphone.	
Formation, Consultation	Formation ou consultation individuelle sur votre site. Formation ouverte régulièrement chez QFS. Toutes les offres sont également disponibles en ligne.	
Documentation	Manuel complet, tutoriel de démarrage, vidéos, blog, fonction de recherche en ligne, aide en ligne par clic droit dans QF-Test, FAQ générale et technique.	