Test Designer 3: HTML5 and Web Application Compatibility

Test Designer 3 supports modern HTML5-based web applications and many of the technologies commonly used to build interactive web applications.

HTML5 introduced new HTML elements, attributes, APIs, and browser behaviors, while the broader modern web platform includes technologies such as JavaScript, CSS, browser APIs, and client-side frameworks.

For additional information about HTML and the web platform, see:

- WHATWG HTML Standard
- MDN HTML Documentation

HTML5 Compatibility

Test Designer supports most commonly used HTML5 features. However, compatibility can depend on how an application implements its UI components and JavaScript event handling.

Modern web applications can trigger actions through multiple mechanisms. For example, an application may use standard browser events, JavaScript event handlers, CSS pseudo-classes, custom components, or framework-specific event processing.

As a result, an action that works with a standard HTML element may behave differently when the same functionality is implemented through a custom component or JavaScript framework.

Click Actions

Test Designer provides multiple methods for interacting with clickable elements, including `click` and `click2`.

In most cases, the standard `click` action is sufficient. However, some applications implement additional JavaScript behavior or custom event handling that may require an alternative click method.

For example, an application may perform additional processing through JavaScript or use CSS pseudo-classes and framework-specific event handlers.

For more information about CSS pseudo-classes, see the MDN CSS pseudo-classes documentation.

If a standard click action does not produce the expected behavior, try the alternative click method or use an action specifically designed for the application's implementation.

File Upload

HTML5 does not prescribe a single implementation pattern for file-upload interactions across all web applications.

Applications may implement file selection using standard HTML elements, JavaScript, custom components, or framework-specific controls.

To accommodate these implementations, Test Designer provides multiple `setFiles` methods. The appropriate method depends on how the application's file-upload component is implemented.

If a file-upload action does not work as expected, verify the underlying HTML element and the application's JavaScript implementation, then select the appropriate `setFiles` method.

Authentication Support

Test Designer supports the following authentication mechanisms:

- Basic Authentication
- NTLM Authentication
- Cookie-Based Authentication
- Digest Authentication
- ADFS with Kerberos Authentication
- Active Directory Authentication
- ADFS with NTLM Authentication when ADFS Extended Protection token checking is disabled
- ADFS with Negotiate Authentication when ADFS Extended Protection token checking is enabled

AIQ Authentication

Authentication to the AIQ application itself can be configured using:

- Internal AIQ credentials
- LDAP authentication, which requires additional configuration

These authentication mechanisms are separate from authentication used by the web applications being automated.

Record and Playback Considerations

Test Designer uses browser instrumentation and injection mechanisms to provide recording and playback capabilities.

Certain application security mechanisms can interfere with these capabilities.

Man-in-the-Middle Detection

Applications or security tools that detect man-in-the-middle (MITM) activity may prevent Test Designer from recording, injecting, or interacting with a page correctly.

If an application uses certificate pinning, traffic inspection, or other mechanisms designed to detect intercepted browser traffic, additional configuration may be required.

Browser Certificate Validation

Test Designer may use certificates as part of its browser instrumentation and traffic interception mechanisms.

Depending on the browser and application configuration, certificate validation can prevent a page from loading correctly or cause the browser to display certificate warnings.

For example, browsers may reject certificates that do not match the certificate expected for a particular domain.

If a page displays an invalid-certificate error when accessed through Test Designer, verify the browser certificate configuration and the application's security policies.

JavaScript Framework Support

Test Designer 3 has been tested with commonly used JavaScript frameworks and web technologies, including:

- React
- AngularJS
- Angular
- jQuery
- ExtJS
- Bootstrap

Support for a framework does not necessarily guarantee compatibility with every application built with that framework.

Framework implementations can vary significantly, and applications may use custom components, third-party libraries, proprietary event handling, or newer framework versions that behave differently from the versions originally tested.

Compatibility Considerations

Although Test Designer is designed to support modern HTML5 applications and commonly used web frameworks, compatibility can depend on several factors, including:

- Browser version
- Framework version
- Application architecture
- Custom JavaScript event handling
- Custom web components
- Security and certificate policies
- Third-party browser extensions
- Application-specific DOM behavior
- Framework-specific event handling

New framework releases, major version changes, forks, and highly customized implementations may require additional validation.

When an application does not behave as expected, first determine whether the behavior is specific to the application's implementation or to the underlying framework.

Support and Feature Requests

Appvance continuously evaluates compatibility with modern web technologies and commonly used frameworks.

If you encounter a web application that does not behave as expected in Test Designer, provide the relevant application details, the Test Designer action being used, and the observed behavior when submitting a support request.

Feature requests and compatibility issues can be submitted through the available Appvance support channels.

Last updated: August 25, 2026

If you have any feedback about this topic, please contact documentation@appvance.ai.

Revision #3
Created 2026-07-31 15:40:17 UTC by Admin
Updated 2026-08-25 06:45:07 UTC by mmora