

This directory contains code and configuration files to run WinUI tests in Helix.

Helix is a cloud hosted test execution environment which is accessed via the Arcade SDK. More details:

- [Arcade](#)
- [Helix](#)

WinUI tests are scheduled in Helix by the Azure DevOps Pipeline: [RunHelixTests.yml](#).

The workflow is as follows:

1. NuGet Restore is called on the packages.config in this directory. This downloads any runtime dependencies that are needed to run tests.
2. PrepareHelixPayload.ps1 is called. This copies the necessary files from various locations into a Helix payload directory. This directory is what will get sent to the Helix machines.
3. RunTestsInHelix.proj is executed. This proj has a dependency on [Microsoft.DotNet.Helix.Sdk](#) which it uses to publish the Helix payload directory and to schedule the Helix Work Items. The WinUI tests are parallelized into multiple Helix Work Items.
4. Each Helix Work Item calls [runtests.cmd](#) with a specific query to pass to [TAEF](#) which runs the tests.
5. If a test is detected to have failed, we run it again, first once, then eight more times if it fails again. If it fails all ten times, we report the test as failed; otherwise, we report it as unreliable, which will show up as a warning, but which will not fail the build. When a test is reported as unreliable, we include the results for each individual run via a JSON string in the original test's errorMessage field.
6. TAEF produces logs in WTT format. Helix is able to process logs in XUnit format. We run [ConvertWttLogToXUnit.ps1](#) to convert the logs into the necessary format.
7. RunTestsInHelix.proj has EnableAzurePipelinesReporter set to true. This allows the XUnit formatted test results to be reported back to the Azure DevOps Pipeline.
8. We process unreliable tests once all tests have been reported by reading the JSON string from the errorMessage field and calling the Azure DevOps REST API to modify the unreliable tests to have sub-results added to the test and to mark the test as "warning", which will enable people to see exactly how the test failed in runs where it did.