

Universal Testing for Console

Overview

Universal Testing is the Microsoft framework for creating and deploying test packages onto just about any device through just about any process. We use it for packaging up all sorts of test resources and sending it into our automated test labs no matter what the source of the content or the engineering system involved.

It involves several parts:

- TESTMD
 - These define a package unit for deployment to the test device. This usually includes the test binaries and any dependent data that it will need to execute.
 - There can also be a hierarchy where one package can depend on another such that packages can be re-used.
- TESTLIST
 - This defines a batch of TESTMD packages that should be executed together.
- TESTPASSES
 - This defines a list of tests via a TESTLIST and a lab environment configuration on which the tests should be run.

These files can either include their child element as they're supposed to (TESTMDs included in TESTLISTs) or they can often include themselves to provide chain structuring (one TESTLIST can reference another TESTLIST).

- TREX
 - This is the legacy configuration system that performed the same job as TESTPASSES, but not in source files.

Configuration

This is a record of the current setup (as of Mar-1-2019) of the console's universal tests. This series of steps was created in conjunction with converting the console's testing from the legacy TREX dispatching mode to the new TESTPASSES dispatching mode for the Source Is Truth initiative (define all testing metadata in source next to the code being tested, instead of in a separate database somewhere else).

1. Have some TestMDs.

- `\onecore\windows\core\console\open\src\host\ut_host\testmd.definition + SOURCES file`

- Generates "Microsoft-Console-Host-UnitTests" TESTMD and package

- Binplaces to `prebuilt\test\<arch>\<fre/chk>\`

1. `Microsoft.Console.Host.UnitTests.testmd`

1. `Microsoft-Console-Host-UnitTests.cab`

1. `Microsoft-Console-Host-UnitTests.man.dsm.xml`

- `\onecore\windows\core\console\open\src\host\ft_host\testmd.definition + SOURCES file`

- Generates "Microsoft-Console-Host-FeatureTests" TESTMD and package

- Binplaces to `prebuilt\test\<arch>\<fre/chk>\`

1. `Microsoft.Console.Host.FeatureTests.testmd`

1. `Microsoft-Console-Host-FeatureTests.cab`

1. `Microsoft-Console-Host-FeatureTests.man.dsm.xml`

- `\onecore\windows\core\console\open\src\buffer\out\ut_textbuffer\testmd.definition + SOURCES file`

- o \onecore\windows\core\console\open\src\testlist\Microsoft.Console.Tests.testlist + SOURCES file

- Includes

- 1. Microsoft.Console.Host.UnitTests.testmd

- 1. Microsoft.Console.Host.FeatureTests.testmd

- 1. Microsoft.Console.TextBuffer.UnitTests.testmd

- 1. Microsoft.Console.Conpty.UnitTests.testmd

- 1. Microsoft.Console.VirtualTerminal.Parser.UnitTests.testmd

- 1. Microsoft.Console.VirtualTerminal.Adapter.UnitTests.testmd

- Binplaces to prebuilt\test\<arch>\<fre/chk>\

- Microsoft.Console.Tests.testlist

- o \onecore\windows\core\console\open\src\testlist\Microsoft.Console.TestLab.Desktop.testlist + SOURCES file

- Includes

- 1. Microsoft.Console.Tests.testlist

- Binplaces to prebuilt\test\<arch>\<fre/chk>\

- Microsoft.Console.TestLab.Desktop.testlist

- Is currently the subject of TREX IDs

- 1. 153251 - TESTLIST based AMD64 Desktop VM testpass (to be offboarded when done)

- 1. 153252 - TESTLIST based X86 Desktop VM testpass (to be offboarded when done)

- \onecore\windows\core\console\open\src\testlist\Microsoft.Console.TestLab.OneCoreUap.testlist + SOURCES file

- Includes

- 1. Microsoft.Console.Tests.testlist

- Binplaces to prebuilt\test\<arch>\<fre/chk>\

- Microsoft.Console.TestLab.OneCoreUAP.testlist

- Is currently the subject of TREX IDs

- 1. 153253 - TESTLIST based AMD64 OneCoreUAP VM testpass (to be offboarded when done)

1. 153254 - TESTLIST based X86 OneCoreUAP VM testpass (to be offboarded when done)

1. Create some TESTPASSES

- o For the existing OneCoreUAP ones...

- Create directory \onecoreuap\testpasses\local\console\

- Create file console_onecoreuap.testpasses

1. Create AMD64 pass

- Name it similarly to the existing name

- Use the environment \$(TESTPASSES_ONECOREUAP)/standard_testenvs/OneCoreUAP-amd64-VM.testenv

- Connect to the testlist
\$(TESTLIST_SEARCH_PATHS)/Microsoft.Console.TestLab.OneCoreUAP.testlist

1. Create X86 pass

- Name it similarly to the existing name

- Use the environment
\$(TESTPASSES_ONECOREUAP)/standard_testenvs/OneCoreUAP-x86-VM.testenv

- Connect to the testlist
\$(TESTLIST_SEARCH_PATHS)/Microsoft.Console.TestLab.OneCoreUAP.testlist

- o For the Desktop ones...

- Create directory \pcshell\testpasses\local\console\

- Create file console_desktop.testpasses

1. Create AMD64 pass

- Name it similarly to the existing name

- Use the environment \$(TESTPASSES_PCSHELL)/standard_testenvs/Enterprise-amd64-VM.testenv

- Connect to the testlist
\$(TESTLIST_SEARCH_PATHS)/Microsoft.Console.TestLab.Desktop.testlist

1. Create X86 pass

- Name it similarly to the existing name

- Use the environment `$(TESTPASSES_PCSHELL)/standard_testenvs/Enterprise-x86-VM.testenv`

- Connect to the testlist
`$(TESTLIST_SEARCH_PATHS)/Microsoft.Console.TestLab.Desktop.testlist`

1. Hook up the TESTPASSES into the official branch TESTPASSES file

- Open up `.\branchconfig\official\rs_onecore_dep_acidev\official_build.testpasses`

1. Add TestpassReferences item targeting
`$(TESTPASSES_ONECOREUAP)/local/console/console_onecoreuap.testpasses`

1. Add TestpassReferences item targeting
`$(TESTPASSES_PCSHELL)/local/console/console_desktop.testpasses`