
Question

[Michael Davidovich](#) · May 18, 2022

Unit Testing Conventions/Best Practices - Where Do They 'Live'?

Continuing on the journey of implementing %UnitTest, @Timothy.Leavitt's Test Coverage package, and automated testing with Jenkins.

My question today: why do we utilize a unit test root directory?

I've been defining packages and classes to write unit tests as I've been developing and I run the tests on the command line or using a routine file that is setup as a debug target which has been working great. I use DebugRunTestCase() to do this (so the classes aren't deleted).

As I am playing with automating this on Jenkins, I'm not understanding exactly why we just don't keep and run them on the server? I may be answering my own question here, but is the idea we don't want the classes living on the server with the application code? Some questions for you all as I learn best practices:

- Are all your unit tests added to .gitignore so they don't get wound up in source code?
- If you do that, how can other developers access the unit tests so they don't write new ones when some already exist?
- Why does the normal RunTestCase() method automatically delete the extracted unit test class files from the folder? Why is that the default?
- When it comes to automated testing (Jenkins specifically) what is the lifecycle? Code is pulled from remote repo, unit test classes extracted into root, all other classes and files compiled on server, run test cases, class files are deleted from root?

Hopefully you can see my biggest gap in understanding seems to be the default functionality of deleting these classes but also needing them on the server to work and develop with. Would another testing framework like JUnit have similar setups that I could learn more from that?

Thanks for any advice!

[#Testing #InterSystems IRIS](#)

Source URL: <https://community.intersystems.com/post/unit-testing-conventionsbest-practices-where-do-they-live>