# Chapter 8: Release Candidate Creation

### Azure Artifacts Manages Release Candidates

Azure Artifacts is an independent product, but it’s used in conjunction with Azure Pipelines. It’s the storage service for the release candidate components produced by the continuous integration build. The application for this article has three deployable components that are built and versioned together.

* Website user interface (UI)
* Database
* Integration tests

The first two can be obvious, but you may be wondering about the integration tests. This deployable package contains test data and testing scripts that are also used to properly set up the TDD environment. You factor it into a separate deployable component because it does need to be deployed to an environment in your pipeline, but it’s not a part of the actual software application that will make its way to the production environment.

Earlier, I stressed how important versioning is in a DevOps pipeline. In **Figure 24**, you inspect the release candidate packages.

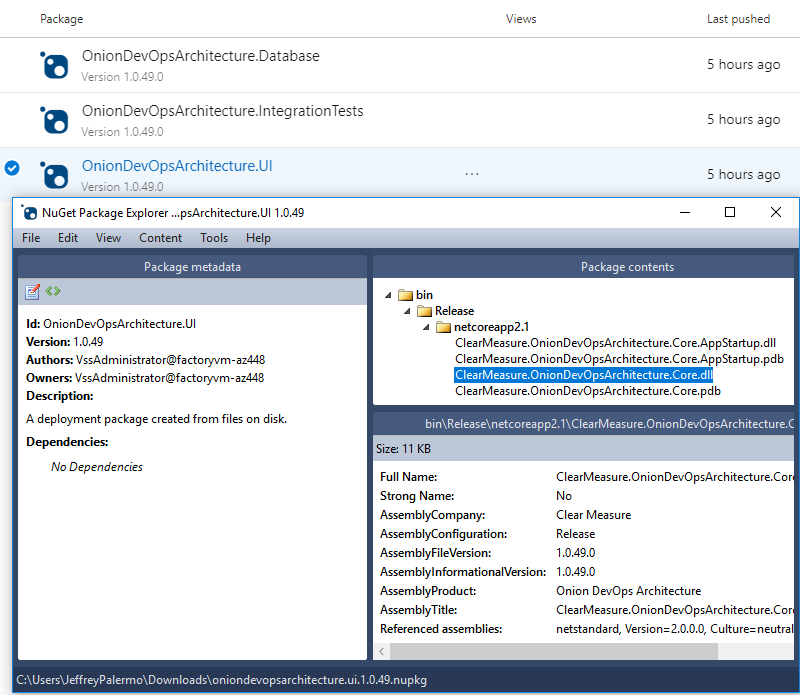


Figure 24: The version of the release candidate is stamped on the NuGet packages as well as every assembly inside.

Because the proper version number is now embedded into every assembly, your code has access to it. Whether you display it at the bottom of the screen or include it with diagnostics telemetry or logs, you’ll use the version number to know whether a problem or bug was on an old version or the current one. Without the version number, you fly blind. Do not try to use date and time stamps to decipher what build you’re working with. Explicitly push the version number into every asset.

Don’t try to use date and time stamps to decipher what build you’re working with. Explicitly push the version number into every asset.