
Article

[John Murray](#) · Mar 7, 2016 1m read

Investigating dependencies

When developing or maintaining Caché Objects applications, or even ones written in plain COS code, it's sometimes useful to investigate dependencies. [Yuzinji](#) is a tool that analyzes the Caché class dictionary plus the COS code within methods and routines, then feeds its results into [Structure101g Studio](#).

By doing this it becomes easy to ask questions such as "Who uses this piece of code?", getting the information in a format that is easy to explore further. For example, I recently analyzed all the packages in the SAMPLES namespace of the 2016.2 Field Test and was interested to observe a dependency on the %apiOBJ routine. Here's a screenshot of what this toolset showed me:

Two classes contain methods that invoke a couple of entrypoints in %apiOBJ, specifically DecomposeStatus^%apiOBJ and DisplayError^%apiOBJ. A cleanup effort might replace these with calls to classmethods of %SYSTEM.Status, i.e. \$System.Status.DecomposeStatus and \$System.Status.DisplayError.

Do you have any favourite tips, tools and techniques to share in this area?

[#Object Data Model](#) [#ObjectScript](#) [#Caché](#)

Source URL: <https://community.intersystems.com/post/investigating-dependencies>