
Discussion

[Evgeny Shvarov](#) · Feb 11, 2019

Tools and Framework Development and Deployment Guidelines . What is The Best Approach?

Hi guys!

Suppose you developed a tool, framework, library on InterSystems ObjectScript for InterSystems Data Platform, share it via [Open Exchange](#) and want people to install it.

What is the best strategy if it is a library and supposed to be called from different namespaces?

Here are the following thoughts:

1. Tool's developer should be able to use globals to read/store data the tool needs.
2. Tool's consumer should be confident, that the thing he/she installs will not harm the application's data.
3. A tool should be callable from any application's namespace.
4. Tool's installation(in general) should not request manual security changes (e.g. grant the write access to IRISLIB) and %YourClass is not an option.
5. It's easy to uninstall the tool - just delete the namespace/database (maybe delete some web-apps too).

In this case I see the following scenario acceptable the most:

1. Every new tool should install itself into a new special namespace and database for your tool. E.g. COMMUNITYLIB. Use %Installer fro then.
2. If you need the tool to be called from other application namespaces, map classes to a %All.

Here are the good examples of this type of tool installations:

[RESTForms](#), [DeepSeeWeb](#), [MONLBL](#) - install itself into new namespaces/databases.

Not perfect examples:

[WebTerminal](#), [ClassExplorer](#) - need write access to the %SYS namespace.

Thoughts?

[#Deployment](#) [#Caché](#) [#InterSystems IRIS](#) [#Open Exchange](#)

Source

URL:<https://community.intersystems.com/post/tools-and-framework-development-and-deployment-guidelines-what-best-approach>