Question
[Robert Cemper](#) · Mar 13, 2023

ECP between IRIS and Caché /Ensemble

Those of you that worked already with ECP might know that it operates not only across
operating systems, processor technologies (big-little-endian), but also -with limits- across versions in C/E.
This cross-version feature also includes IRIS (verified for 2020.1) and C/E (2018.3)

There are of course limits and you are not as free as in pure C/E only environments.
It is not a good idea to share code across this type of ECP connection.

While .MAC and .INT routines **may** work *eventually*. Class Code will definitely not.
2 Examples:
Storage of a %Serial class:

- C/E : `<Type> %Library.CacheSerialState</Type>`
- IRIS: `<Type>%Storage.Serial</Type>`

Storage of a %Persisten Class:

- C/E : `<Type> %Library.CacheStorage</Type>`
- IRIS: `<Type>%Storage.Persistent</Type>`

The consequences:
You need to port your classes from C/E to IRIS
You need only to map the referred GLOBALS ! (or all Globals)
But this is a rather simple and well-documented exercise.

In General:
Mapping only Globals over ECP should be the preferred approach anyhow since they are the moving parts.
Code (Routines and Classes) does not change in production systems so often on the fly that an immediate update
is required.
So you reduce the load on your ECP connection and win performance by better using your available bandwidth.

With IRIS the different inheritance is just an additional push to separate code from data.
(which makes sense anyhow. eg. backups)

This approach allows you to make limited use of some IRIS features
and reduces time pressure before you run a full migration to IRIS.

It follows the idea of a micro-service, allowing to split a big BANG into smaller bang, bang, bang steps.

#ECP #Globals #Caché #Ensemble #InterSystems IRIS