
Article

[Robert Cemper](#) · Mar 5, 2021 3m read

Using ECP across IRIS and Caché

Migration from Caché to IRIS can be quite a challenge if your code is grown over many years and probably not so clean structured as you may like it. So you face the need to check your migrated code against some reference data. A few samples might not be a problem, but some hundred GB of data for testing might be.

A possible step could be to have your fresh code in IRIS but leave your huge datastore on Caché and connect both environments over ECP. I have created a demo project that gives you the opportunity to try this based on 2 Docker images with IRIS and with Caché connected over ECP.

Attention:

- Both Docker images require a personal license for MultiServer to enable ECP
- The default Community License doesn't allow ECP and can't be used for Caché.
As a customer with a support contract, you may get loan licenses directly from WRC.

Scenario:

Caché acts as ECP Server while IRIS acts as ECP Client

In IRIS you have a namespace SAMPLES.

Globals are in remote database SAMPLES on Caché

Routines (and Classes) are in database USER

The classes were just migrated by drag/drop from Caché Studio to Iris Studio

Data-Globals are in remote database SAMPLES on Caché

This setup allows you to have local data in namespace USER

and remote data in namespace SAMPLES and run your test queries or other exercises.

Installation:

- Get the external IPV4 address of the machine that runs your docker environment (example =10.10.1.99)
This is required to establish access between both containers
- Download [CrossECP-Caché from OEX](#)
- Copy your (loan) license key into cache.key
- From the download directory run:
docker-compose up -d --build and you are done with Caché.
It uses -p 41773:1972 for the Caché super server and -p 42773:57772 for the webserver
Your actual directory is mapped to /external to allow file exchange with docker environment
- Next Download [CrossECP-IRIS from OEX](#)
- Copy your (loan) license key into iris.key
- From the download directory run:
docker-compose up -d --build
It uses these port mappings -p 45773:1972 -p 46773:52773 -p 47773:53773
Your actual directory is mapped to /external to allow file exchange with docker environment
- To complete installation feed your Docker host IP address (10.10.1.99) and start operation run:
docker-compose exec iris iris session iris initECP

```
Server status 1 Not Connected
Continue anyway ? (nNyY) [Y]: Y
Enter Host-IP-Adress of Docker (nn.nn.nn.nn) [192.168.0.6]: 10.10.1.99
Connect to ECP sever on 10.10.1.99 now ? (nNyY) [Y]: Y
Server status 5 Normal
```

This last step is just for your comfort. Of course, you can do this also from SMP
by System > Configuration > ECP Settings > ECP Data Servers

Now you are ready for testing.

[GitHub](#)

[#ECP #Caché #InterSystems IRIS](#)

Source URL: <https://community.intersystems.com/post/using-ecp-across-iris-and-cach%C3%A9>