**Xtclang XDK environment build and PaaS install**

Start clean! Note this runbook is targeted at MacOS and can be tweaked for linux…we’re not addressing windows install for the moment.

**Dependencies**  
Homebrew (<https://brew.sh/>)   
Java 21 (needs to be java 21 for now as kotlin compilation errors produced later in build against anything more recent)  
Xtclang PaaS (<https://github.com/xtclang/platform>)  
Xtclang XDK (<https://github.com/xtclang/xvm#installation>)

**Runbook**

Install homebrew from the link above and follow the ‘Install Homebrew’ command into a terminal window and let it run. If this is your first time note the brew comments at the end of the install prompting you to add brew to your profile.

Install java 21 in a terminal window  
brew install openjdk@21

Set JAVA\_HOME (adjust for local specialness!)  
echo ‘[export="JAVA\_HOME=/opt/homebrew/Cellar/openjdk@21/21.0.5/libexec/openjdk.jdk/Contents/Home](https://encoded-592c9deb-987b-4562-aa3c-9fa3d37d83e9.uri/mailto%3aexport%3d%2522JAVA_HOME%3d%2fopt%2fhomebrew%2fCellar%2fopenjdk%4021%2f21.0.5%2flibexec%2fopenjdk.jdk%2fContents%2fHome)"’ >> ~/.zshrc

## Install XVM build from source

Create a folder ~/xtclang/xvm   
Go [here](https://github.com/xtclang/xvm) and download the repo from the A green rectangle with white text

Description automatically generated button via the “Download ZIP”  
Unzip this download and copy the contents into ~/xtclang/xvm  
Open a terminal and cd to ~/xtclang/xvm and issue  
./gradlew installDist

From the terminal issue (note path edit required )  
echo ‘export XDK\_HOME="/Users/**[YOURUSERACCOUNT]**/xtclang/xvm/xdk/build/install/xdk"’ >> ~/.zshrc

echo ‘export PATH="$XDK\_HOME/bin:$PATH"’ >> ~/.zshrc

Close your terminal window and re-open to refresh your updated profile and check for it with   
echo $XDK\_HOME

Now open your profile to verify   
nano ~/.zshrc

It should show at least these entries if you ran the homebrew instructions above and have installed openjdk21 from Homebrew (ymmv!) and added the $XDK\_HOME variable to your profile.  
  
export PATH="/opt/homebrew/opt/openjdk@21/bin:$PATH"  
[export JAVA\_HOME=/opt/homebrew/Cellar/openjdk@21/21.0.5/libexec/openjdk.jdk/Contents/Home](https://encoded-592c9deb-987b-4562-aa3c-9fa3d37d83e9.uri/mailto%3aexport%2520JAVA_HOME%3d%2fopt%2fhomebrew%2fCellar%2fopenjdk%4021%2f21.0.5%2flibexec%2fopenjdk.jdk%2fContents%2fHome)export XDK\_HOME=”/Users/**[YOURUSERACCOUNT]**/xtclang/xvm/xdk/build/install/xdk”  
export PATH="$XDK\_HOME/bin:$PATH"

### The XDK requires the applications xcc and xec. They are easily installed like this

cd $XDK\_HOME/bin

./cfg\_macos.sh

## PaaS Install

Go [here](https://github.com/xtclang/platform) and download the repo from the A green rectangle with white text

Description automatically generatedbutton via the “Download ZIP”   
Under ~/xtclang create a folder platform  
Unzip this download and copy the contents into ~/xtclang/platform  
At ~/xtclang create a folder called xqiz.it  
At ~/xtclang/xqiz.it create a folder called platform  
At ~/xtclang/xqiz.it create a folder called users  
In ~/xtclang/xqiz.it/platform create a file called port-forwarding.conf  
Open a terminal to edit this file and paste the following text into it  
  
rdr pass on lo0 inet proto tcp from any to self port 80 -> 127.0.0.1 port 8080  
rdr pass on lo0 inet proto tcp from any to self port 443 -> 127.0.0.1 port 8090

Save and close the file  
Issue this command to redirect incoming http\* traffic to the PaaS . You will be prompted for your pwd  
sudo pfctl -evf ~/xtclang/xqiz.it/platform/port-forwarding.conf  
  
If this gives an error reopen the port-forwarding.conf file and before the first rdr entry place a hard return, save, close and try again and it should resolve.

From your terminal issue  
ping xtc-platform.localhost.xqiz.it

which should resolve and show successful packet responses from 127.0.0.1

**Resolve PaaS dependencies and build**~~Go~~ [~~here~~](https://github.com/Boardmad/depends) ~~and download the repo from the A green rectangle with white text

Description automatically generatedbutton via the “Download ZIP”  
Unzip the downloaded zip and copy the package-lock.json file into ~/xtclang/platform/platformUI/gui~~

In a terminal from within ~/xtclang/platform issue  
brew install gradle node yarn

Change directory to ~/xtclang/platform/platformUI/gui and issue  
npm install

Then do (this step may prove to be redundant – leave in for completeness now)  
npm install -g @quasar/cli

Change directory to ~/xtclang/platform and issue  
gradle clean build

This step will take up to a minute to compile for the first time but will be far quicker in subsequent builds

If this is NOT the first build and you receive warnings about platformDB instances being out of sync with existing builds try the following.

cd to ~/xtclang/xqiz.it and delete the accounts folder  
cd to ~/xtclang/xqiz.it/platform and delete everything **EXCEPT** port-forwarding.conf  
cd to ~/xtclang/xqiz.it/users and delete anything in there (it may be empty)

\*We need to document the quasar version update to package.json somewhere? Should we hard code it here?

Change directory back to ~/xtclang/platform and try again  
gradle clean build

**Start the PaaS**

To start the PaaS GUI navigate to ~/xtclang/platform and issue  
xec -L lib/ lib/kernel.xtc **[password]**

Where **[password]** is replaced with your preferred admin account password

First run will create self-signed certs and local security keys for the PaaS and can be opened by visiting [here](https://xtc-platform.localhost.xqiz.it/). Note the security errors you will be notified you are visiting an unsecured page. This is known behaviour can be ignored for now.

Once you have verified that the PaaS is up and have completed testing, to shut down the server cleanly, issue this from a new terminal window/tab

curl -k -b cookies.txt -L -i -w '\n' -X POST <https://xtc-platform.localhost.xqiz.it/host/shutdown>

**Examples for hosting on Ecstasy PaaS**Go [here](https://github.com/xtclang/examples) and download the repo from the A green rectangle with white text

Description automatically generatedbutton via the “Download ZIP”

Under ~/xtclang create a folder examples and unzip and copy the downloaded repo contents into it

From a terminal navigate to ~/xtclang/examples/banking and issue

gradle build

This will build the banking example \*.x files

Now start the PaaS GUI by navigating to ~/xtclang/platform and issue  
xec -L lib/ lib/kernel.xtc [password]

Where “[password]” is replaced with your preferred admin account password

A screenshot of a phone

Description automatically generatedNow navigate [here](https://xtc-platform.localhost.xqiz.it/) accepting the security warnings to open the PaaS UI

Login using the button at the top right of the screen admin / [password you set above] \*Note as of 17/12/24 the login on the PaaS UI appears to be broken and returns a ‘Login Failed’ error element in the UI but still allows the admin account to login.  
Select the modules menu item on the left of the PaaS UI then click the ‘install new module’ icon from the PaaS UI

A brown rectangle with white text

Description automatically generated In the popup that opens click the plus sign and navigate to ~/xtclang/examples/banking/server/build and select the \*.xtc files and click open.

Next click the grid with edit pencil icon to the left of the delete icon which says ‘register application’ when you hover over it

A screenshot of a white background

Description automatically generated

Then give your instance of the application a name (test2) and click register.

A close-up of a computer screen

Description automatically generated

Now click the small green ‘run’ icon to the left of the ‘delete’ icon. This will ‘run’ this instance of your application and the red inactive cloud icon should change to a green ‘Active’. You can now click on the urls to open an instance of the web ui for this example application.

WOOT!