
Article

[Zhong Li](#) · Apr 12, 2019 11m read

Build HealthShare Docker Images for Demo Purposes on a Windows Laptop

1. Purpose

This is a 10-minute simple step-by-step guide on how to quickly set up various flavors of HealthShare docker containers from scratch on a Win10 laptop.

For example, we can build a couple of HealthShare "global edition vs UK Edition" demos as shown below.

There are a couple of frequently asked questions from HealthShare colleagues and partners:

- "I am no Docker guy, but is there a quick way to build various flavors of HealthShare containers simply for demo/PoC/dev/training or troubleshooting purpose?"
- "I just can't make "Docker for Windows" work on my Win10 laptop - how did you make that work? What's the simplest/easiest way to play with HealthShare containers on my old Windows laptop?"

The truth is I am not Docker specialist either - I wish I had time for it. I am using an old laptop, and I haven't even tried "Docker for Windows" yet.

I happen to be using "[Docker Toolbox](#) for Windows" for the past 18 months to play with ICM, HealthShare and IRIS containers, and it worked well so far on a densely populated old laptop. I didn't notice any major show-stoppers yet - please let me know if you happen to encounter one.

2. Scope

The main components used in this quick guide are simply:

- [Docker Toolbox](#) for Windows - it is a legacy application now but downloadable and works well
- HealthShare 2018.1 for Centos(RHEL64) release
- [WebTerminal](#) - so you can run HealthShare/Cache Terminal in a web browser
- Windows 10 Pro laptop

In this guide we will:

- Build a HealthShare 2018.1 Docker image from scratch
- Run a HealthShare container as a "global edition" demo.
- Run another HealthShare container as a "UK Edition" demo.
- See what's next.

A note about HealthShare here: "Docker Toolbox" actually uses VirtualBox as a Docker Machine, and all HealthShare containers in this guide are actually running as Centos applications on VirtualBox which in turns runs on a Win10 Pro OS.

Other than that the same step-by-step guide (except step 4.1 below) would work well for any other Docker platforms, such as Docker Desktop for Windows, Docker for MacOS and Linux etc.

Disclaimer:

- Sure, "10-minute" is a figurative speech; in reality it might take up to 1-2 hours to run the Dockerfile to build your 1st HealthShare docker image, and to configure various flavors of containers. After then it should just take a few seconds to run it.
- Docker Toolbox becomes a legacy application now.

3. Prerequisite

It would need at least 40G+ disk space on your Windows laptop.

There is no other prerequisite than you have to be familiar with InterSystems HealthShare products, their installation and configurations on common Linux platforms.

No "Docker Desktop for Windows" or Hyper-V is needed. Docker knowledge or experiences (of any kind) would be helpful but not mandatory. We will avoid any deep dive in Docker itself.

4. Steps from the scratch

4.1 Install "Docker Toolbox for Windows"

["Docker Toolbox for Windows" can be download from here.](#)

Please follow [the screen-by-screen installation guide](#) to install "Docker Toolbox for Windows" application on your Windows laptop.

Please run Step 3 "Verify your installation" of the above link to make sure the installation is successful. You can choose all default settings as a quick start. "Optional: Add shared directories" is not needed in this guide, but may be helpful for your further plays.

After the above is done, We can start "Docker Quickstart Terminal" then run the following. We can see a VM with hostname "Default" having been created as a Docker Machine:

```
zhongli@UKE7450ZLEE MINGW64 /h
$ docker-machine ls
NAME ACTIVE DRIVER STATE URL SWARM DOCKER ERRORS
default * virtualbox Running tcp://192.168.99.100:2376 v18.09.3
```

4.2 Increase Docker machine's disk size

Currently (by today April 2019) HealthShare is not really designed for Docker yet, so it's still a relatively large footprint in disk size - will be a 6G+ for a standard base installation.

So first we will increase the Docker machine VM's disk size to e.g. 50G.

1. Close all "Docker Quickstart Terminal" windows.
2. Find and edit this file: C:/Program Files/Docker Toolbox/start.sh, by adding a line in bold red as below, then save it

... ..

```
STEP="Checking if machine $VM exists"
if [ $VMEEXISTSCODE -eq 1 ]; then
```

... ..

... ..

```
PROXYENV="$PROXYENV --engine-env NOPROXY=$NOPROXY"
fi
"${DOCKERMACHINE}" create -d virtualbox --virtualbox-disk-size "50000" $PROXYENV "${VM}"
fi
```

```
STEP="Checking status on $VM"... ..
```

Note, now you can simply remove the newly create docker machine "default" by typing docker-machine rm default in Docker QuickStart Terminal, then close and restart the Terminal - it will create a new "default" docker machine with a 50G vmdk disk.

4.3 Create a Dockerfile for HealthShare

Using e.g. Notepad++ to create a Docker file - you can simply copy across the sample content below, edit & save it with a file name of "Dockerfile"

```
# This Docker manifest file builds a container with:
# - sshd running (linux containers don't usually have it)
# - HealthShare 2018.1 for RHEL x64 and
# - it handles container PID 1 via ccontainermain which offers various flags
#
# build the new image with e.g.
# $ docker build --force-rm --no-cache -t hs:18.01 .
#--

# pull from this repository
# note that if you don't have the distribution you're after it will be automatically
# downloaded from Docker central hub repository (you'll have to create a user there)

FROM tutum/centos:latest

MAINTAINER zhongli <Zhong.Li@intersystems.com>

# setup variables for the HealthShare
#
ENV TMPINSTALLDIR=/tmp/distrib
ENV HSDIST="HealthShare-2018.1-ExchangeInsightIndex-b7718-lnxrhx64.tar"

# vars for HealthShare installation
ENV ISCPACKAGEINSTANCENAME="HEALTHSHARE"
ENV ISCPACKAGEINSTALLDIR="/usr/healthshare"
ENV ISCPACKAGEINITIALSECURITY="Normal"
ENV ISCPACKAGEUNICODE="Y"
```

```
ENV ISCPACKAGEUSERPASSWORD="SYS"
ENV ISCPACKAGECSPSYSTEMPASSWORD="SYS"
ENV WEBTERMINALDIST="WebTerminal-v4.8.0.xml"

# HealthShare distribution_
# set-up and install HealthShare from distribtmp dir
RUN mkdir ${TMPINSTALLDIR}
WORKDIR ${TMPINSTALLDIR}
COPY ${HSDIST} ${TMPINSTALLDIR}
COPY ${WEBTERMINALDIST} ${TMPINSTALLDIR}

# update OS + dependencies & run HealthShare silent install
RUN yum -y update /
&& yum -y install which tar bzip2 hostname net-tools wget java /
&& yum -y clean all /
&& ln -sf /etc/locatime /usr/share/zoneinfo/Europe/London

RUN tar xvfC ${HSDIST} ${TMPINSTALLDIR}
RUN ./HealthShare-*/cinstallsilent
RUN ccontrol stop $ISCPACKAGEINSTANCENAME quietly
COPY cache.key $ISCPACKAGEINSTALLDIR/mgr/

RUN ccontrol start $ISCPACKAGEINSTANCENAME /
&& printf "SuperUser.h${ISCPACKAGEUSERPASSWORD}.h" /
| csession $ISCPACKAGEINSTANCENAME -U USER
"##class(%SYSTEM.OBJ).Load("/${TMPINSTALLDIR}/${WEBTERMINALDIST}"/,"cdk/")"

RUN rm -rf ${TMPINSTALLDIR}/*
RUN ccontrol stop $ISCPACKAGEINSTANCENAME quietly

# TCP sockets that can be accessed if user wants to (see 'docker run -p' flag)
EXPOSE 56772 56773 57772 57773 57774 1972 22 80 443

# Caché container main process PID 1 (https://github.com/zrml/ccontainermain)
WORKDIR /
ADD ccontainermain .

ENTRYPOINT ["/ccontainermain", "-cconsole"]

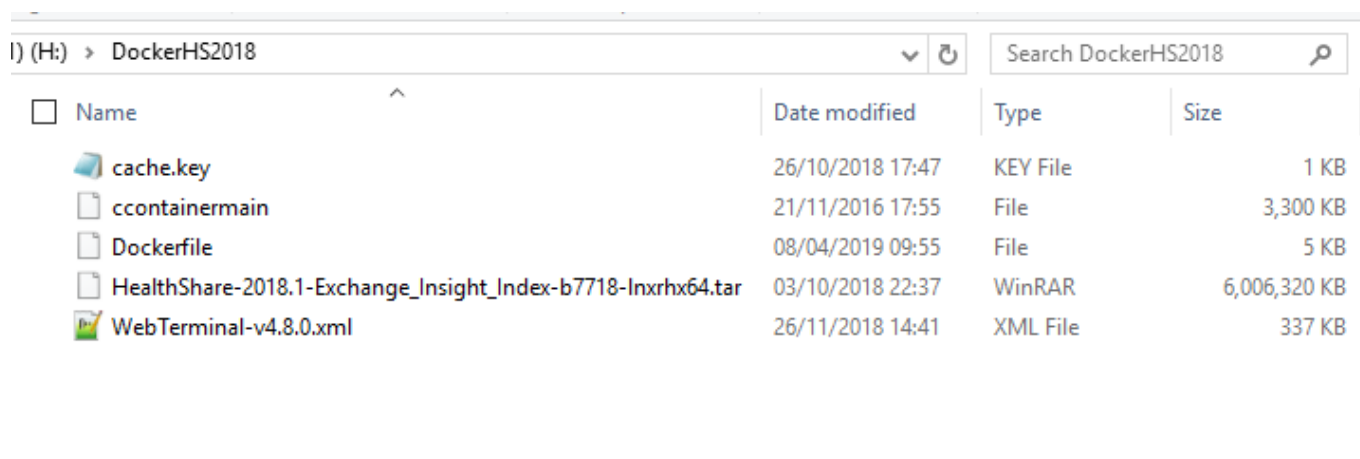
# run via:
# docker run -d -p 57772:57772 -p 1972:1972 -e ROOTPASS="linux" --name HSTEST hs:18.01
-i=HEALTHSHARE
#
# more options & explanations
# $ docker run -d // detached in the background; accessed only via network
# --privileged // only for kernel <= 3.16 like CentOS 6 & 7; it gives us root privileges to tune the kernel etc.
# -h <hostname> // you can specify a host name
# -p 57772:57772 // TCP socket port mapping as hostexternal:containerinternal
# -p 0.0.0.0:2222:22 // this means allow 2222 to be accessed from any ip on this host and map it to port 22
in the container
# -e ROOTPASS="linux" // -e for env var; tutum/centos extension for root pwd definition
# <dockerimageid> // see docker images to fetch the right name & tag or id
# // after the Docker image id, we can specify all the flags supported by 'ccontainermain'
# // see this page for more info https://github.com/zrml/ccontainermain
# -i=HealthShare // this is the Caché instance name
# -xstart=/run.sh // eXecute another service at startup time
# // run.sh starts sshd (part of tutum centos container)
```

// for more info see <https://docs.docker.com/reference/run/>

4.4 Check the working directory

Now we can list the complete working directory needed to build our first HealthShare docker image.

For example, my working directory is H:\HS20181, shown as below:



Name	Date modified	Type	Size
cache.key	26/10/2018 17:47	KEY File	1 KB
ccontainermain	21/11/2016 17:55	File	3,300 KB
Dockerfile	08/04/2019 09:55	File	5 KB
HealthShare-2018.1-Exchange_Insight_Index-b7718-lnxrhx64.tar	03/10/2018 22:37	WinRAR	6,006,320 KB
WebTerminal-v4.8.0.xml	26/11/2018 14:41	XML File	337 KB

We should have the following files within the working directory:

1. cache.key - A valid HealthShare key file
2. ccontainermain - Can be downloaded from here: <https://github.com/zrml/ccontainermain>; there is a very good description on why we need it.
3. Dockerfile - the text file we just created/saved as above.
4. HealthShare-2018.1-ExchangeInsightIndex-b7718-lnxrhx64.tar - you can download from WRC Download page this HealthShare 2018.1 installation file for REHL/Centos x64 (note: you need to unzip it to tar format here)
5. Webterminal-v4.8.0.xml - WebTerminal component can be download from InterSystems Developer Community: <https://community.intersystems.com/post/cach%C3%A9-webterminal-v4-release>

The above are all you need to build a HealthShare 2018.1 docker imagine.

4.5 Build our first HealthShare Docker image

Now we simply start "Docker Quick Start Terminal", then run this command with the Terminal window, to build our first HealthShare Docker image labeled as "hs:18.01"

```
zhongli@UKE7450ZLEE MINGW64 /h/DockerHS2018  
$ docker build --force-rm --no-cache -t hs:18.01 .
```

Note: It may take quite a while, up to 30-45 minute, for the first run. If there is an issue, you can simply re-run the above command and it will carry on from where it failed - the Dockerfile is still fairly bare and not fit for exception handlings, but the underlining build tool is fairly robust.

If the build is successful, the you can inspect the images that we just build by running "docker image ls"

```
zhongli@UKE7450ZLEE MINGW64 /h
$ docker image ls
REPOSITORY TAG IMAGE ID CREATED SIZE
hs 18.01 1d7f1216edc4 5 weeks ago 21.1GB
tutum/centos latest 99a633ad346f 3 years ago 297MB
```

4.6 Run our 1st HealthShare Container as a "global edition"

Now we can simply run our first HealthShare container application "HSTEST", by using a line of command below within the same Docker Terminal window:

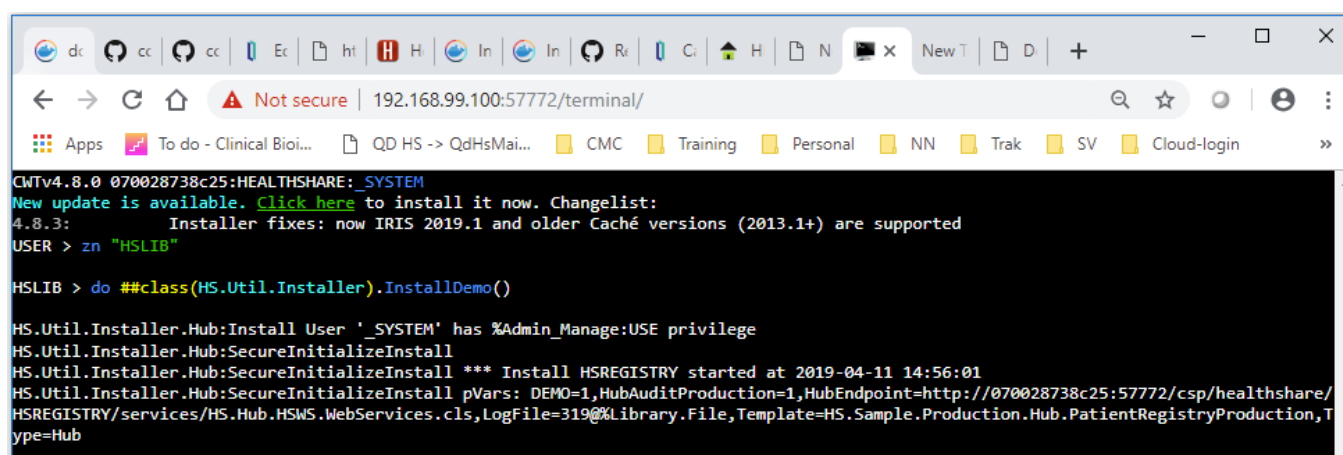
```
docker run -d -p 57772:57772 -p 1972:1972 -e ROOTPASS="linux" --name HSTEST hs:18.01 -i=HEALTHSHARE
```

This command will create a HealthShare container listening on port 1972 and 57772 of our Docker machine "Default" (default local IP 192.168.99.100 - can be configured):

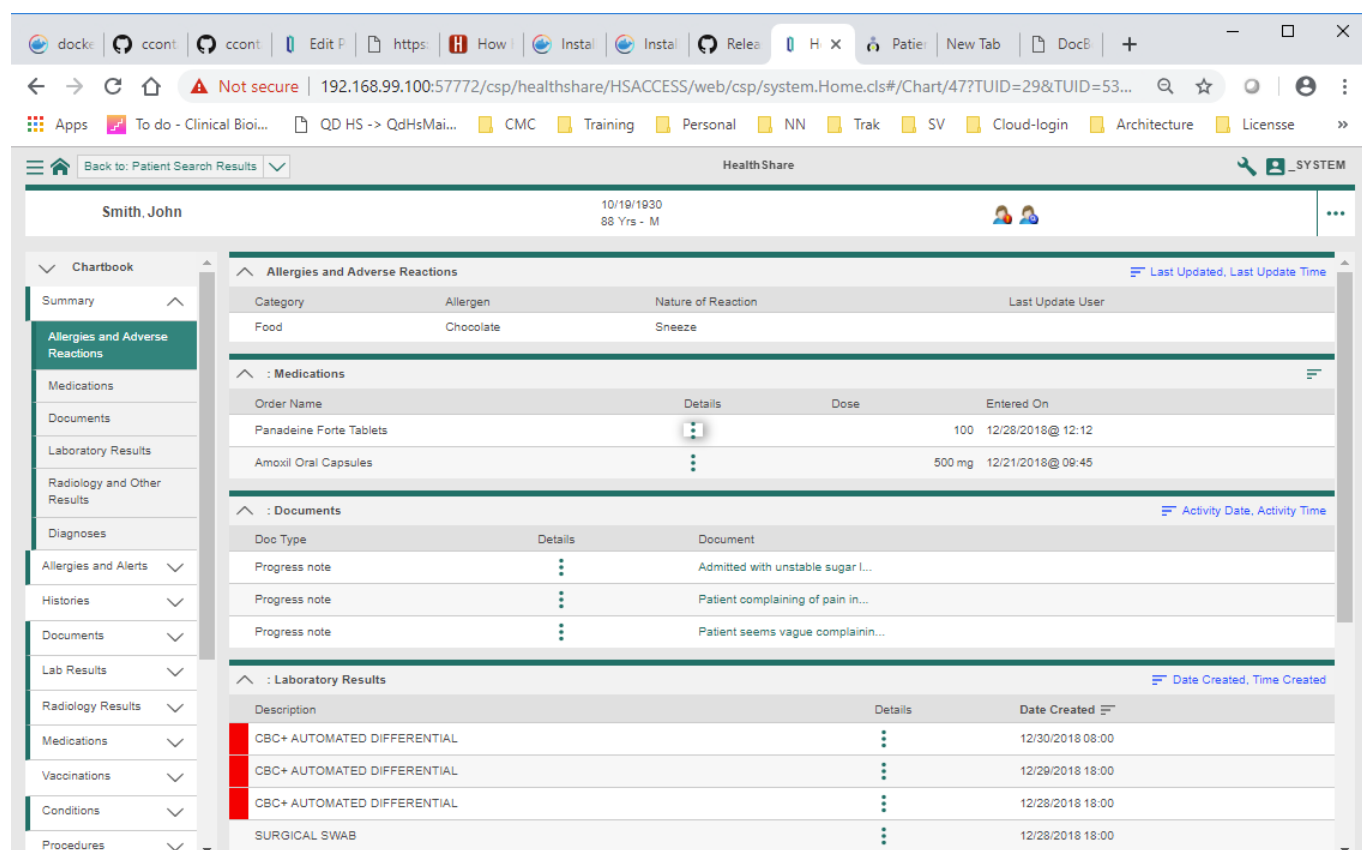
```
zhongli@UKE7450ZLEE MINGW64 /h
$ Docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
070028738c25 hs:18.01 "/ccontainermain -cc" 5 weeks ago Up 2 days 22/tcp, 80/tcp, 443/tcp,
0.0.0.0:1972->1972/tcp, 56772-56773/tcp, 57773-57774/tcp, 0.0.0.0:57772->57772/tcp HSTEST
```

Now we can certainly start to try all sort of HealthShare operations that we normally do for HealthShare demo/PoC/troubleshooting and all kind of configurations.

For example, we can quickly invoke the WebTerminal to run HealthShare's "InstallDemo()" utility, to set up those few common test patients for a quick issue etc:



To see the following HS components were created, and to invoke the CV with a test patient:



4.7 Run our 2nd HealthShare container as a "UK Edition"

By using the same image "HS:18.01", we can also create another HealthShare container, then configure it to be a "UK Edition 2018.1":

```
docker run -d -p 57792:57772 -p 1992:1972 -e ROOTPASS="linux" --name HSUK hs:18.01 -i=HEALTHSHARE
```

Then we can see container "HSUK" is running on ports 1992 and 57792 of the Docker machine:

```
$ Docker ps -a
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
68a78be76fac hs:18.01 "/ccontainermain -cc" 2 days ago Up 2 days 80/tcp, 443/tcp, 56772-56773/tcp, 57773-57774/tcp, 0.0.0.0:58892->22/tcp, 0.0.0.0:1992->1972/tcp, 0.0.0.0:57792->57772/tcp HSUK
070028738c25 hs:18.01 "/ccontainermain -cc" 5 weeks ago Up 3 days 22/tcp, 80/tcp, 443/tcp, 0.0.0.0:1972->1972/tcp, 56772-56773/tcp, 57773-57774/tcp, 0.0.0.0:57772->57772/tcp HSTEST
```

Now we can deploy our "HealthShare UK Edition 2018.1" into this Container instance, then use it for e.g. Testing and Training purpose, such as Clinical Viewer V2 training - actually that's exactly what we did in our "HealthShare Engineering Week 2019" event.

So we can see a UK flavor of the HS 2018.1 CV2:

Patient Search

Not secure | 192.168.99.100:57792/csp/healthshare/hsaccess/HS.Local.HSUK.UI.Patient...

Patient Search Results

Select one or more matching patients. Only select more than one matching patient if you are certain they are the same person.

Identifiers	Name	Gender	DOB	Address Rank/InfoTypes
<input type="checkbox"/>	Jackson, Alice Matilda	F	24 Sep 1935	7
<input checked="" type="checkbox"/> NHS No 485 836 4968	Jackson, Tilly	F	24 Sep 1935	7

[Back to: Patient Find](#) [View Selected](#)

Reason For Access

NHS Number485 836 4968

NameJACKSON,Tilly

Date of Birth24-Sep-1935

GenderFemale

☐ Claim a legitimate relationship with this patient

☒ Request one-time access

Please give a reason you need to access the patient record in the absense of patient consent and/or legitimate relationship.

Authorised by*

☒ Patient

☐ Other (please specify)

Authorisation Type*

☐ Verbal

☒ Written

☐ Emergency

Authorisation Reason*

Emergency

Cancel

Continue

Any inappropriate breach of patient confidentiality will be a matter for disciplinary and potentially legal and/or professional proceedings. If in doubt, speak to your manager or privacy officer.

The screenshot displays the HealthShare web application interface. At the top, the browser address bar shows the URL `192.168.99.100:57792/csp/health`. The application header includes the patient's name, **Jackson, Tilly**, and various demographic details: **Born 24-Sep-1935 (83y)**, **Gender Female**, and **NHS No. 485 836 4968**. The address is listed as **24 Ridley Road, Lincoln, LN3 6BT**. A sidebar on the left contains a 'Chartbook' menu with options like Summary, Allergies and Adverse Reactions (selected), Medications, Diagnoses, Documents, Laboratory Results, Radiology and Other Results, Allergies and Alerts, Histories, Documents, Lab Results, Radiology Results, Vaccinations, Conditions, Procedures, Discharge Summaries, Physical Exams, Plans, Encounters, Referrals, Appointments, Care Team, Programmes, and SCR. The main content area shows several expandable sections: 'Allergies and Adverse Reactions' with columns for Category, Allergen, Nature of Reaction, and Last Update User; ': Medications' with columns for Order Name, Details, Dose, and Entered On; 'Diagnoses' with columns for Diagnosis Type, Description, ICD Code, Status, Care Provider, Date, and Last Updated; ': Documents' with columns for Doc Type, Details, Activity Date, and Document; ': Laboratory Results' with columns for Description, Details, and Date Created; and ': Radiology and Other Results' with columns for Description, Details, and Collection Date. Each section has a sort icon and a link to view details.

5 What's next

Docker container is indeed well fit for running many various flavors of HealthShare applications that are logically separated, and have a well consolidated foot print on the hard disk, apart from other well known DevOps advantages.

By comparing with VMs, its footprint is much smaller. An example is our "HealthShare UK Demo" which used to take about 90G to copy across without counting various snapshots. By using Docker images it would merely a few more Gs (on top of a 21G base HealthShare image) to accommodate all the UK specific classes, config and demo data. I also like the fact it only takes a few seconds to run from a pre-built image.

By comparing with direct installations, Docker certainly has the distinct advantages of "Portability", which is a common productivity boost. It would be slightly tricky (although do-able) to copy across a configured local instance to another laptop and make it running - but we do need quite a few hours to "hack" it through. A Docker image would save the day.

Next, we can simply commit various flavors of our above configured HealthShare containers into Docker images and exchange with other colleagues for further Dev, Testing, Demo and Troubleshooting purposes.

6 Acknowledgement

Thanks to Colin Fallon who worked out a good Dockerfile on his MacBook that really gave a boost here to finish off this long-due posting and make them all running better on my Docker Toolbox for Windows, in the hope to save a bit effort (no matter how little it is) for anyone who is interested in running HealthShare in Docker containers.

7 Caveats

1. HealthShare is not officially released for Docker Container yet. IRIS and IRIS Health are officially released as Docker certified applications, so we don't have to build their images from scratch anymore.
2. Docker Toolbox for Windows becomes a legacy application. You can simply use "Docker Desktop for Windows" to build your own HealthShare images if it is working on your PC.

[#Docker](#) [#HealthShare](#)

Source

URL: <https://community.intersystems.com/post/build-healthshare-docker-images-demo-purposes-windows-laptop>