

HDX-ADM-100 DXClient for Beginners

HCLSoftware U

Creating a new generation of experts

Edition : November 2023

Herbert Hilhorst – herbert.hilhorst@hcl-software.com

Table of Contents

Author(s)..... 3

Introduction..... 3

Prerequisites..... 4

Lab Overview 5

Part 1: Download and extract DXClient Image 6

Part 2: Deploy DXClient Docker 9

Part 3: Configure and Test DXClient 10

Part 4: Use with different artifacts 14

Conclusion 15

Resources 16

Legal statements 17

Disclaimers..... 18

Author(s)

This document was created by the following Subject Matter Experts:



Herbert Hilhorst
Company: HCL

Bio

Herbert Hilhorst is an HCL Digital Experience (DX) Technical Advisor at HCL.

Contact: Herbert.hilhorst@hcl-software.com

Introduction

This hands-on lab introduces you to the DXClient tool of HCL Digital Experience (DX) that you may use to stage any DX resources.

In this DX administrator lab, you play the role of a DX Administrator and experience how easy it is to set up allowing you to manage your DX server and stage DX resources easily.

Prerequisites

1. Completion of [HDX-INTRO](#) and [HDX-BU-100](#) courses, including the labs
2. Docker installed and set up. You can get it from <https://www.docker.com/>.
3. Alternatively, you may use Podman, <https://podman.io/>, and use alias `docker=podman` to keep the same command lines. This is not officially supported, but should work.
4. DXConnect should be configured on your DX server. This is the case for container based deployment, but it may need to be configured on traditional installations. In that case follow these steps https://opensource.hcltechsw.com/digital-experience/latest/extend_dx/development_tools/dxclient/dxconnect/.

You will be using the following user IDs and passwords:

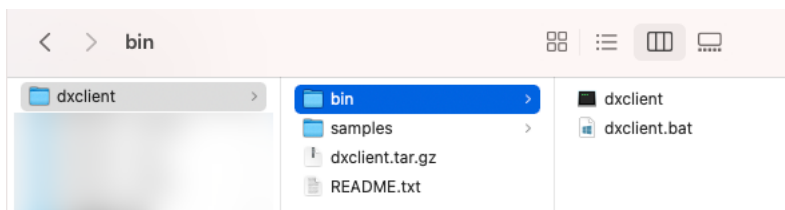
Purpose	User	Password
SoFy Login	Your official email id	Your password
SoFy Solution Console Login	sol-admin	pass
DX Login	wpsadmin	wpsadmin

Lab Overview

In this lab there are several parts to get you started with the DXClient tool. These are shortly introduced now.

Part 1: Download and extract DXClient Image

You will download the DXClient image and extract it.



Part 2: Deploy DXClient Docker

Then deploy it to your workstation as a Docker image.

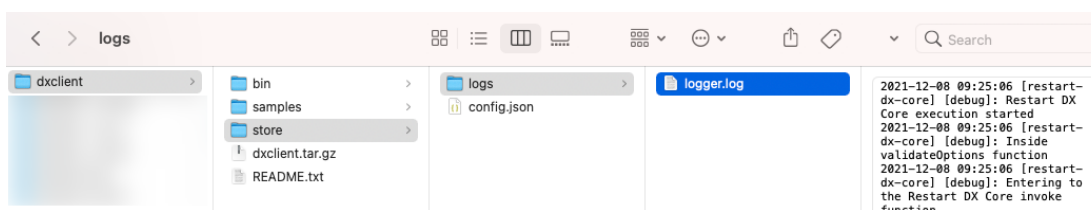
```

Hilhorst-MacBook-Pro:dxclient hilhorst2$ docker load < dxclient.tar.gz
bc7bdf0ec1b9: Loading layer 104.6MB/104.6MB
d7ecef9dcc97: Loading layer 20.48kB/20.48kB
7fde3abed81a: Loading layer 86.88MB/86.88MB
6b8713523c59: Loading layer 125.9MB/125.9MB
4ecfb5b023ff: Loading layer 229MB/229MB
a562b5253ac4: Loading layer 5.655MB/5.655MB
6b167782a43f: Loading layer 5.654MB/5.654MB
11a38b00325e: Loading layer 50.63MB/50.63MB
Loaded image: hcl/dx/client:latest
Hilhorst-MacBook-Pro:dxclient hilhorst2$

```

Part 3: Configure and Test DXClient

Then, configure it to work with your DX server with global settings and test it.



Part 4: Use for different artifacts

And in this last part, you will learn how to further configure this to work in other places more easily, if you are working with different artifacts.

Part 1: Download and extract DXClient Image

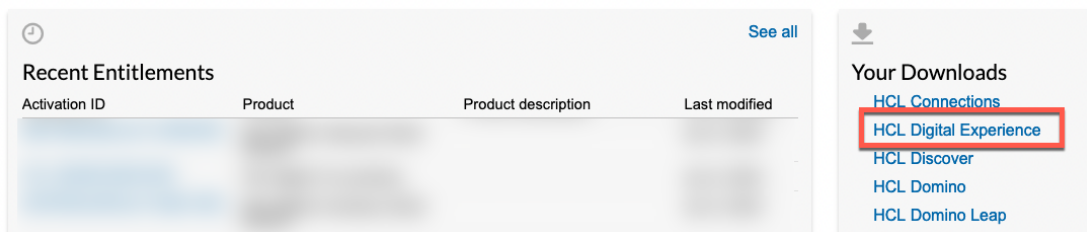
In this part, you will first download the DXClient Image and then extract it. You may use the DX download steps document for additional information on getting access to the resources using: <https://hclsw.co/dx-download-steps>. In this part, you may then follow the instructions below to get the DXClient image.

1. Log on to the License & Delivery Portal site: <https://hclsw.co/flexnet>. Under **Your Downloads**, scroll down and select **HCL Digital Experience**.

HCLSoftware



License & Download Portal



2. In the list search for **HCL_Portal_8.5_9.0_9.5_CFs** and click to access.

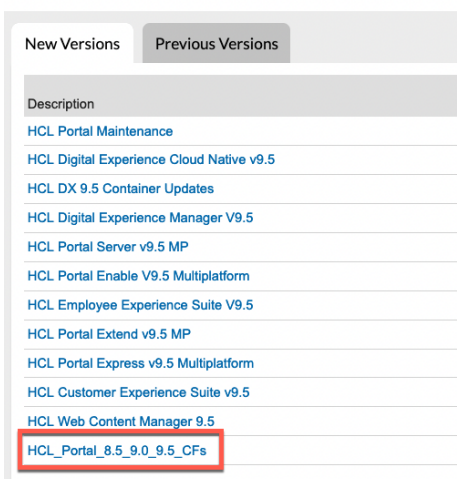
HCLSoftware



Download Packages

The Download Packages page lists new and archived versions of dov

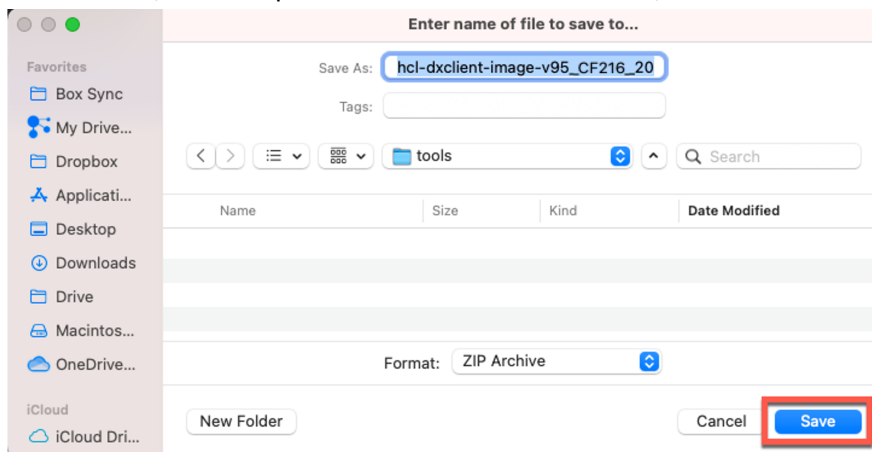
HCL Digital Experience



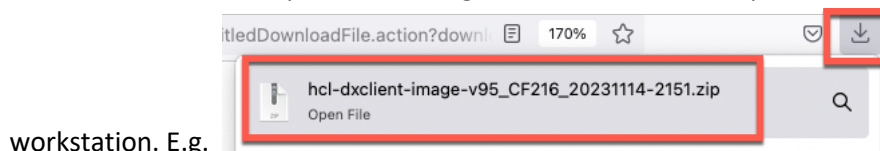
- Then scroll and search for the version of the **HCL_DX_CFXXX Client Image** that matches your server and click on Client Image **hcl-dxclient-image-v95_CFXXX_XXXXXXXX-XXXX.zip** to download it, in this case it is shown for CF216.

<input type="checkbox"/> +	HCL DX 9.5 CF215 Client Image	205.9MB	Oct 06, 2023	hcl-dxclient-image-v95_CF215_20231004-1332.zip
<input type="checkbox"/> +	HCL DX 9.5 CF216	7.97GB	Nov 16, 2023	hcl-dx-kubernetes-v95-CF216.zip
<input type="checkbox"/> +	HCL DX 9.5 CF216 Client	4.09MB	Nov 16, 2023	hcl-dxclient-v95_CF216_20231114-2151.zip
<input type="checkbox"/> +	HCL DX 9.5 CF216 Client Image	215.08MB	Nov 16, 2023	hcl-dxclient-image-v95_CF216_20231114-2151.zip
<input type="checkbox"/> +	HCL_DX_CF214_Express_Update	1.54GB	Aug 17, 2023	HCL-DX-CF214_Express_Update.zip

- It asks you to select a place to save it. Select a working directory from which you wish to use the DXClient, for example a tools folder as shown here, and click **Save**.

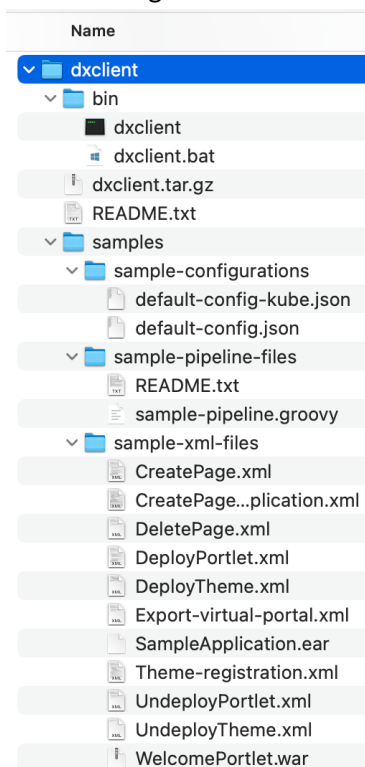


- If you are upgrading your client, you may want to rename your old dxclient folder, e.g. to 'dxclient cf215'. Then open the file to get it extracted. This may be different on your



workstation. E.g.

- This should give you this folder structure with dxclient on the top. Have a look around the different files and folders. Under the **bin**, you will find the executable file for Linux/Mac and Windows. In the **samples** folder, you will find several sample files to get started more easily, and the **README.txt** has a detailed documentation on the DXClient tool. It also has the **dxclient.tar.gz** which is the DXClient container.



You have successfully downloaded and extracted the DXClient image and are ready to deploy it.

Part 2: Deploy DXClient Docker

In this part, you will deploy it to your workstation as a Docker image. You need to have Docker installed and set up in your workstation. There are several places that gives you access to download and deploy Docker. For example from <https://docs.docker.com/get-docker/>. You may also use any container runtime that implements OCI Runtime Specification, for example Podman: <https://podman.io/>. In that case, run this command *export CONTAINER_RUNTIME=<YOUR_CONTAINER_RUNTIME>*. For Podman: *export CONTAINER_RUNTIME=podman* to keep the same command lines. Once set up, follow these steps to deploy the DXClient Docker image.

1. Open a shell or Command Window first. Then go to the path where you extracted the image run the following command, **docker load < dxclient.tar.gz**, which should load it and give an output like this:

```
herberthilhorst dxclient % docker load < dxclient.tar.gz
ad5362eb8973: Loading layer [=====>] 94.52MB/94.52MB
63020b64d0f9: Loading layer [=====>] 84.02MB/84.02MB
35e6f29da85c: Loading layer [=====>] 129.3MB/129.3MB
ad6a78756844: Loading layer [=====>] 271.9MB/271.9MB
36c5f3914beb: Loading layer [=====>] 11.31MB/11.31MB
b821c8c7e84d: Loading layer [=====>] 11.31MB/11.31MB
1715faef4a1d: Loading layer [=====>] 52.58MB/52.58MB
Loaded image: hcl/dx/client:v95_CF216_20231114-2138
herberthilhorst dxclient %
```

Congratulations! You have successfully deployed DXClient on your local workstation.

Part 3: Configure and Test DXClient

Then configure it to work with your DX server, set the global settings and test it. To check if it works, use DXClient with one of the command options. Ensure you run this in a place where you want your configuration to be setup. This can be in the dxclient directory where you extracted the client, but also in other areas, e.g. inside each Script Application directory you wish to be able to deploy. Each time you run it in a different place, you need to set up your configuration again (that you may copy from your original).

1. Go to the working directory where you want to use the DXClient. Here we show you to run it from the dxclient directory, you extracted before. Run **dxclient -V**, which should give you the current version. In this case, 1.25.0, which is the version of CF216. See the Help Center for details: https://opensource.hcltechsw.com/digital-experience/latest/extend_dx/development_tools/dxclient/#verifying-your-dxclient-installation.

```
herberthilhorst@ [redacted] dxclient % ./bin/dxclient -V
1.25.0
```

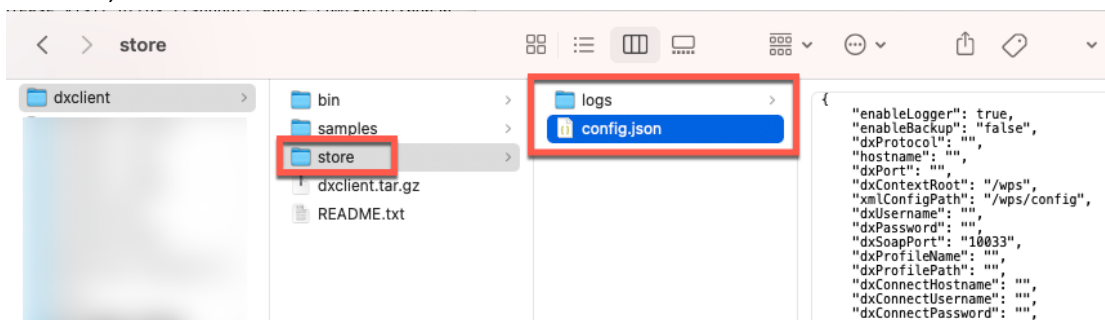
2. If you are using a Mac with an M1 processor, you may see this message.

```
herberthilhorst@ [redacted] dxclient % ./bin/dxclient -V
WARNING: The requested image's platform (linux/amd64) does not match the detected host platform (linux/arm64/v8) and no specific platform was requested
1.17.0
```

3. You may solve this using an additional parameter to run docker: **--platform linux/amd64**. You may add this option in your bin/dxclient file for the **\$CONTAINER_RUNTIME** entry, here showing in line 78:

```
77
78 $TTY_FLAG -e TZ=$Timezone -v "$(pwd)"/$VOLUME_DIR:$DXCLIENT/store:Z --platform linux/amd64 --network="host"
79
```

4. When you run this DXClient the first time on a directory, it will create a store folder under that directory. This folder contains a logs subfolder where all logs will be stored and a config.json file. The config.json allows you to configure all the global settings for the DXClient, such as the server hostname.



5. Now try the help function, run **bin/dxclient -h**. This gives you instructions on the available commands.

```
herberthilhorst@ dxclient % ./bin/dxclient -h
Usage: dxclient [options] [command]

DXClient is a CLI tool to manage tasks such as deploying portlets, script applications and themes easily from source development environments to target HCL DX deployments. This tool is capable of taking artifacts developed locally and deploying them into DX servers deployed on supported on-premise platforms in standalone, cluster or farm topologies and Kubernetes. Kubernetes platforms are supported for HCL Digital Experience 9.5 CF192 and higher only.

Common command arguments that are listed in `config.json` can be pre-configured via this file. The common arguments could also be provided through command line. If so, it will override the values in `config.json` and execute. A sample configuration file 'default-config-kube.json' for Kubernetes environments is available under the 'samples/sample-configurations' folder for reference.

Options:
  -V, --version                output the version number
  -h, --help                   display help for command

Commands:
```

6. You may optionally try it now, using the version compare tool, documented here https://opensource.hcltechsw.com/digital-experience/latest/extend_dx/development_tools/dxclient/dxclient_artifact_types/versioncompat/. The version-compat allows you to check if your DX server is the same level as your DXClient. Check again the help documentation first with **bin/dxclient version-compat -h**.

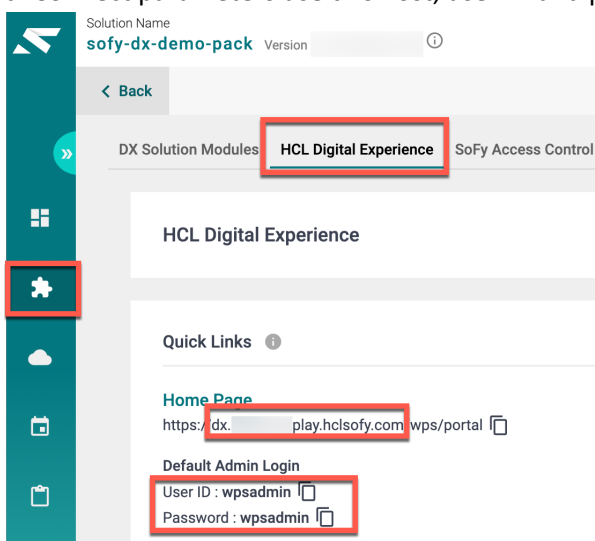
```
herberthilhorst@ dxclient % ./bin/dxclient version-compat -h
Usage: dxclient version-compat [options]

- Show version compatibility details between DX Core and DXClient.

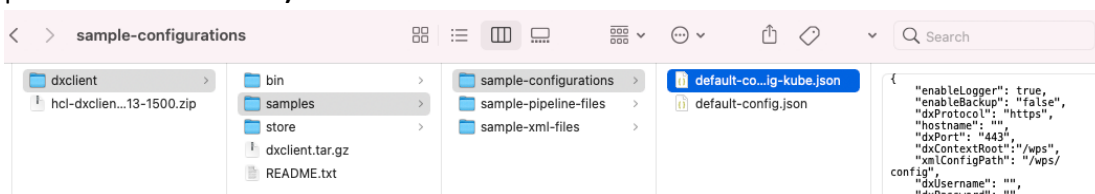
Options:
  --hostname <value>          Host name of the DX Core server (default: "")
  --dxConnectPort <value>     Port number of the cw_profile, for any Kubernetes Environment
                              default port is 443 (default: "10202")
  --dxConnectProtocol <value> Protocol of the cw_profile (default: "https")
  --dxConnectUsername <value> Username for the cw_profile (default: "")
  --dxConnectPassword <value> Password for the cw_profile (default: "")
  -h, --help                  display help for command

herberthilhorst@ dxclient %
```

7. You notice you need to set several parameters to make it work with your SoFy instance: the **hostname**, **dxUsername**, **dxPassword**, **dxConnectUsername** and **dxConnectPasword**. You can find these in your **Solution Console**, under the **Solution Content – HCL Digital Experience** tab, you have the Quick Links session that gives you this. Both the dx and dxConnect parameters use this host, user ID and password.



- Then you refer to the sample config.json files under the **samples/sample-configurations** folder, that have these parameters set up with reference values for your environment for both a container (default-config-kube.json) and non-container based (default-config.json) deployment. In this lab, you will connect it to your HCL SoFy instance, so you will reuse the container-based example. It helps you to find the values for the missing parameters: **dxProfileName wp_profile** or **dxProfilePath /opt/HCL/wp_profile** (as only of these parameters are needed) and **dxConnectPort 443**.



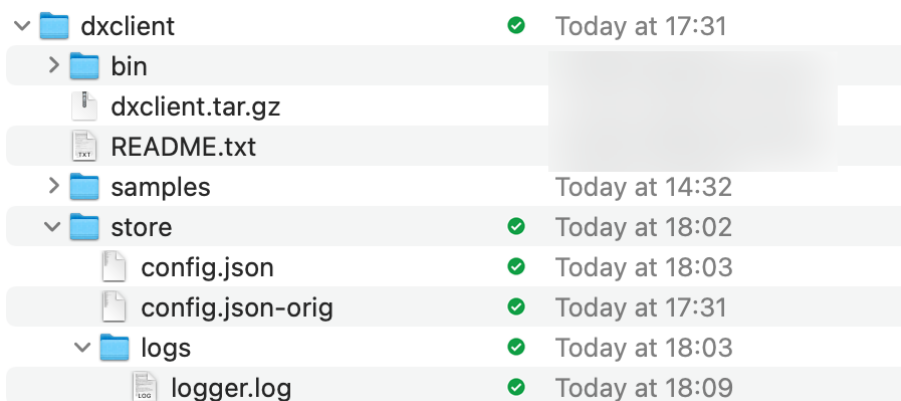
- Now you can run the command file with these parameters, in this case (update with your host) **bin/dxclient version-compat -hostname dx.sbx0000.play.hclsofy.com -dxConnectPort 443 -dxConnectUsername wpsadmin -dxConnectPassword wpsadmin**

```
herberthilhorst@ dxclient % ./bin/dxclient version-compat -hostname dx.sbx0000.play.hclsofy.com -dxConnectPort 443 -dxConnectUsername wpsadmin -dxConnectPassword wpsadmin

Warning: The CF versions of both DXclient and DX Core are not identical and there is the possibility for incompatibilities.
DX Version: CF215
Current DXClient Version: 1.25.0
Recommended DXClient Version: 1.24.0

herberthilhorst@ dxclient %
```

- Notice that in this case, the DX server was on an older version. Hence it recommends you to use the same DXClient version. The process is logged under the **logs/logger.log** file. If an output is generated, you may find that under the **store/outputFiles** directory. The version-compat does not generate any outputs.



11. To avoid having to add all these parameters to each command line, you may set them in the `dxclient/store/config.json` file. Apply this for example to the `hostname`, `dxConnectPort`, `dxConnectUsername` and `dxConnectPassword`, `dxUsername`, `dxPassword` and `dxProfileName` entries. Then **Save** it:

```

1  { "name": "config.json",
2    "desc": "The attributes in this file are used for configuration purpose and those must not be deleted.",
3    "dxProtocol": "",
4    "dxConnectProtocol": "https",
5    "hostname": "dx.██████████.play.hclsofy.com",
6    "dxPort": "",
7    "dxSoapPort": "10033",
8    "dxConnectPort": "443",
9    "dxContextRoot": "/wps",
10   "contenthandlerPath": "/wps/mycontenthandler",
11   "projectContext": "",
12   "virtualPortalContext": "",
13   "xmlConfigPath": "/wps/config",
14   "damAPIPort": "",
15   "ringAPIPort": "",
16   "damAPIVersion": "v1",
17   "ringAPIVersion": "v1",
18   "dxConnectHostname": "",
19   "dxConnectUsername": "wpsadmin",
20   "dxConnectPassword": "wpsadmin",
21   "dxUsername": "wpsadmin",
22   "dxPassword": "wpsadmin",
23   "dxProfileName": "wp_profile",
24   "dxProfilePath": "",
25   "dxWASUsername": "",
26   "dxWASPassword": "",
27   "dashboards": []

```

12. Now you may run the `restart-dx-core` without these globally set parameters. Run it with `bin/dxclient version-compat`. You notice it runs without the additional parameters. In case you wish to override any of the parameters in the `config.json`, just add them in your command line.

```

herberthilhorst@██████████ dxclient % ./bin/dxclient version-compat

Warning: The CF versions of both DXClient and DX Core are not identical and there is the possibility for incompatibilities.
DX Version: CF215
Current DXClient Version: 1.25.0
Recommended DXClient Version: 1.24.0

herberthilhorst@██████████ dxclient % █

```

13. You are ready to use it now! You may use the samples in the `samples/sample-xml-files` directory to try this out as well. In other labs, you may learn how to apply it for specific resources.

You have successfully configured and tested the DXClient !

Part 4: Use with different artifacts

And in this last part, you will learn how to further configure this to work in other places more easily if you are working with different artifacts.

While you can always run the DXClient from the same place and configure the parameters, you may also run it where you create your development artifacts, like a Script Application.

1. If you wish to run DXClient from different directories places, you need to add this bin directory to your PATH variable. You can do this using the export PATH call, where you add this DXClient bin directory to your path. To easily get your current working directory, you may use **pwd** to get it.

```
herberthilhorst@ dxclient % pwd
/Users/herberthilhorst/Tools/dxclient
```

2. Copy this path and then use it to add the dxclient/bin directory to your path
export PATH=<working-directory>/bin:\$PATH

```
Hilhorst-MacBook-Pro:dxclient hilhorst2$ export PATH=/Users/herberthilhorst/tools/dxclient/bin:$PATH
Hilhorst-MacBook-Pro:dxclient hilhorst2$
```

3. You can check that is worked, using the **echo \$PATH** command. Notice that on some environments, like a Mac, you may need to perform additional tasks to make this work permanently. You may want to create or update the **~/.bash_profile** file and add this line into the file: **export PATH=<working directory>/dxclient/bin:\$PATH**. Then you can set it more easily using **source ~/.bash_profile**.
4. When you run it now in a different directory, it will create again the store folder with the logs and the config.json file. You may copy the config.json file you configured earlier over this new one created and then further update it for your artifact.

You have successfully configured your DXClient to easily work for different artifacts.

Conclusion

Using this lab tutorial, you have learned how to deploy the DXClient tool as a Docker image, set the global configuration, optionally tested it to restart your server and to make it easily work for different artifacts. You may use the Help Center https://opensource.hcltechsw.com/digital-experience/latest/extend_dx/development_tools/dxclient/ and other labs in the DX courses where specific artifacts, like themes, Script Applications, etc. are managed, to learn more on this tool.

Resources

Refer to the following resources to learn more:

HCL Digital Experience Home - <https://hclsw.co/dx>

HCL Digital Experience on HCL Solutions Factory (SoFy) - <https://hclsofy.com/>

HCL Software - <https://hclsw.co/software>

HCL Product Support - <https://hclsw.co/product-support>

HCL DX Product Documentation - <https://hclsw.co/dx-product-documentation>

HCL DX Support Q&A Forum - <https://hclsw.co/dx-support-forum>

HCL DX Video Playlist on YouTube - <https://hclsw.co/dx-video-playlist>

HCL DX Product Ideas - <https://hclsw.co/dx-ideas>

HCL DX Product Demos - <https://hclsw.co/dx-product-demo>

HCL DX Did You Know? Videos - <https://hclsw.co/dx-dyk-videos>

HCL DX GitHub - <https://hclsw.co/dx-github>

Legal statements

This edition applies to version 9.5, release 208 of HCL Digital Experience and to all subsequent releases and modifications until otherwise indicated in new editions.

When you send information to HCL Technologies Ltd., you grant HCL Technologies Ltd. a nonexclusive right to use or distribute the information in any way it believes appropriate without incurring any obligation to you.

©2023 Copyright HCL Technologies Ltd and others. All rights reserved.

Note to U.S. Government Users — Documentation related to restricted rights — Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with HCL Technologies Ltd.

Disclaimers

This report is subject to the HCL Terms of Use (<https://www.hcl.com/terms-of-use>) and the following disclaimers:

The information contained in this report is provided for informational purposes only. While efforts were made to verify the completeness and accuracy of the information contained in this publication, it is provided AS IS without warranty of any kind, express or implied, including but not limited to the implied warranties of merchantability, non-infringement, and fitness for a particular purpose. In addition, this information is based on HCL's current product plans and strategy, which are subject to change by HCL without notice. HCL shall not be responsible for any direct, indirect, incidental, consequential, special or other damages arising out of the use of, or otherwise related to, this report or any other materials. Nothing contained in this publication is intended to, nor shall have the effect of, creating any warranties or representations from HCL or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of HCL software.

References in this report to HCL products, programs, or services do not imply that they will be available in all countries in which HCL operates. Product release dates and/or capabilities referenced in this presentation may change at any time at HCL's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way. The underlying database used to support these reports is refreshed on a weekly basis. Discrepancies found between reports generated using this web tool and other HCL documentation sources may or may not be attributed to different publish and refresh cycles for this tool and other sources. Nothing contained in this report is intended to, nor shall have the effect of, stating,

or implying that any activities undertaken by you will result in any specific sales, revenue growth, savings or other results. You assume sole responsibility for any results you obtain or decisions you make as a result of this report. Notwithstanding the HCL Terms of Use (<https://www.hcl.com/terms-of-use>), users of this site are permitted to copy and save the reports generated from this tool for such users own internal business purpose. No other use shall be permitted.