HCL Digital Experience

HDX-ADM-100 DXClient for Beginners

Edition : February 2025

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

Table of Contents

Author(s)

This document was created by the following Subject Matter Experts:

Herbert Hilhorst
Company:
HCLSoftware

Bio

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

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 deploy and stage any DX resources.

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

Prerequisites

1. Completion of HDX-INTRO and HDX-BU-100 courses, including the labs
2. DXClient may be installed using JavaScript. This is the recommended way. You need to have Node.js installed. Check out what version of Node.js you should install in https://help.hcl-software.com/digital-experience/9.5/latest/extend_dx/development_tools/dxclient/#installing-dxclient-using-the-native-javascript-package-from-hcl-software-portal. Download and install instructions are here: https://nodejs.org/en/download/package-manager.
3. DXClient may also run using OCI-based runtimes such as Docker or Podman. In this lab, optional instructions are provided using Docker, so you need to have this installed and set up. You can get it from https://www.docker.com/. Alternatively, you may use Podman, https://podman.io/, and use *export CONTAINER_RUNTIME=podman* to keep the same command lines. See details and more installation options in: https://help.hcl-software.com/digital-experience/9.5/latest/extend_dx/development_tools/dxclient/#installing-dxclient-using-the-container-package.
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://help.hcl-software.com/digital-experience/9.5/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
DXConnect/Config Wizard 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.

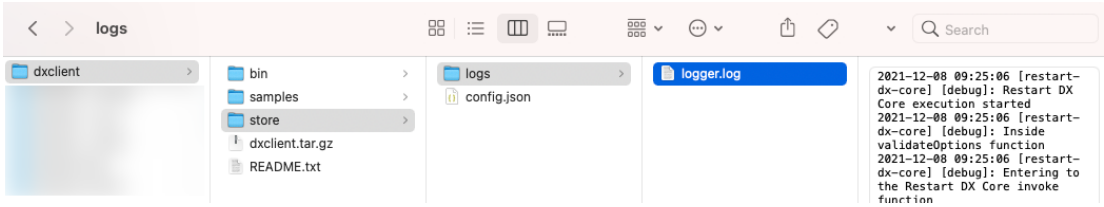
Part 1: Install DXClient from NpmJS Registry

First install the DXClient from the NpmJS registry, using the npm install @hcl-software/dxclient command.

```
s % sudo npm install -g @hcl-software/dxclient
Password:

added 206 packages in 2s

23 packages are looking for funding
  run `npm fund` for details
```

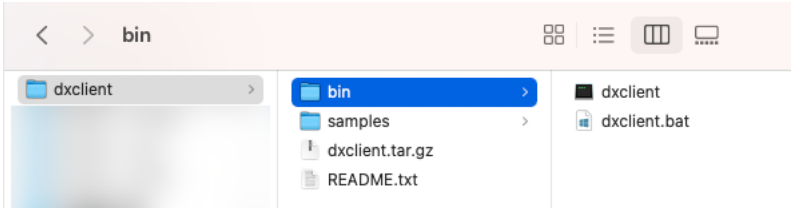
Part 2: Configure and Test DXClient

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



Optional Part 3: Download, extract and deploy DXClient Container Package

Use this optional part, if you prefer to use the DXClient container package. You will learn how to download and extract it.



And then deploy it to your workstation as a Docker image.

```
dxclient — -bash — 80×24
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$
```

Optionally, you may learn how to further configure this to work in other places more easily, if you are working with different artifacts.

Part 1: Install DXClient from NpmJS Registry

First install the DXClient globally from the NpmJS registry, using the npm install -g @hcl-software/dxclient command.

1. First ensure you have installed a recent version of node.js. You may find details to download and install from https://nodejs.org/en/download/package-manager. Open a shell or command and run the command **node -v**. The version should be supported one, as described in the prerequisites part of this lab.

```
% node -v
v20.14.0
```

2. Then run the command **npm install -g @hcl-software/dxclient** command to install the latest version of DXClient globally. If you do not have enough rights, you may need to use **sudo npm install -g @hcl-software/dxclient** instead. To update later again to the latest version, just run this command again. Notice you may also install several local versions, without the -g option. You may ignore any warnings.

```
s % sudo npm install -g @hcl-software/dxclient
Password:

added 206 packages in 2s

23 packages are looking for funding
  run `npm fund` for details
```

You have successfully installed the DXClient.

Part 2: 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. For example, you may create a specific dxclient directory. Then run **dxclient -V**.

```
% dxclient -V
2024-07-31 10:38:48 : Directory created at: /Users/herberthilhorst/Library/Application Support/dxclient/
LICENSE AGREEMENT FOR NON-WARRANTED PROGRAMS:
DXCLIENT ("DXClient" or "Tool")

DXClient is a command line tool that presents an interface to automation and continuous integration and de
 HCL Digital Experience Programs.
```

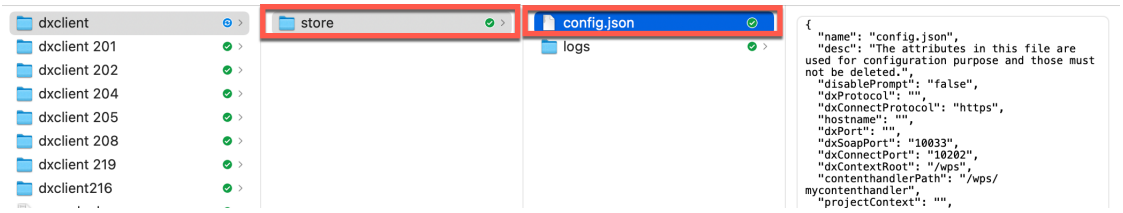
2. This will ask you to accept the license. Read it first, and if you are willing to accept, select **I accept the terms of license & agreement** and hit enter.

```
Version Date: May 27 2024



            HCLSoftware is a division of HCL Technologies Limited
        With registered address at 806 Siddharth 96 Nehru Place New Delhi 110019
                www.hcltechsw.com



( ) I do not accept the terms of license & agreement
(x) I accept the terms of license & agreement
```

3. And then give you the current version. In this case, CF221.0.0, which is the version of DX CF221.

```
2024-07-31 10:42:40 : Terms of license & agreement accepted.
221.0.0
```


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. Notice that if you run DXClient later in a different directory, it will create this under that directory again and you may need to reconfigure the config.json again.



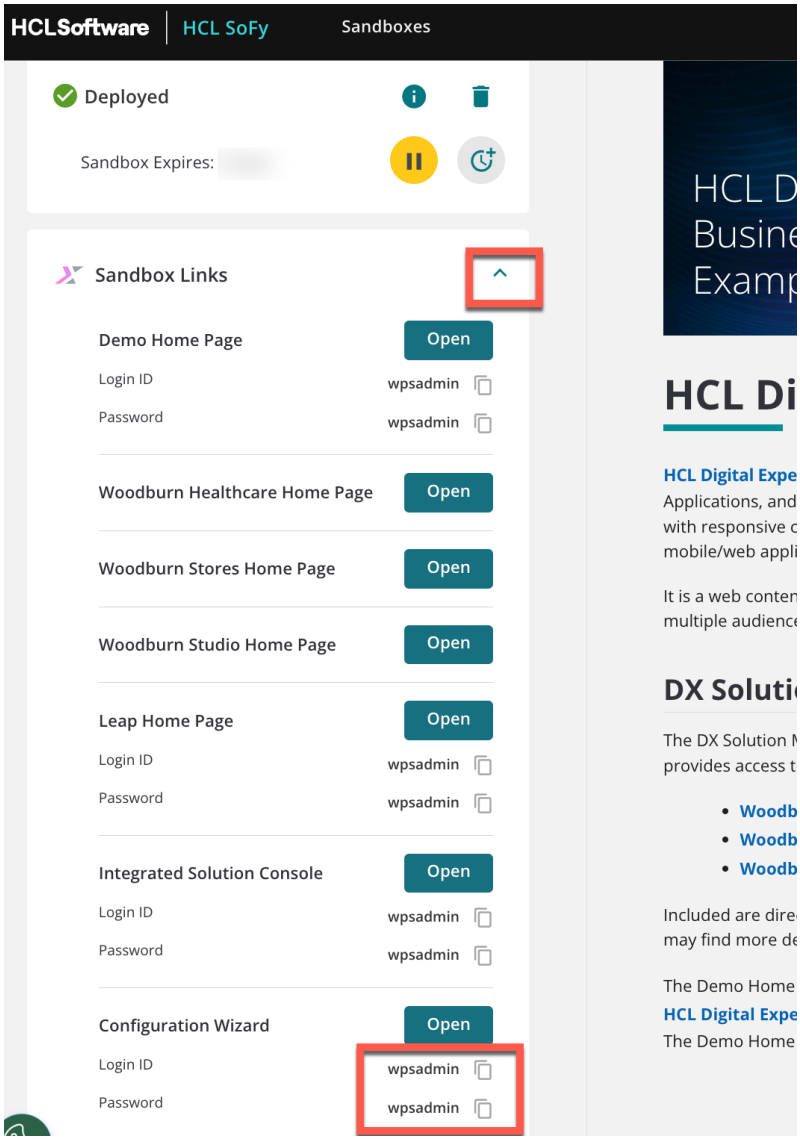
5. Now try the help function, run **dxclient -h**. This gives you instructions on the available commands.

```
dxclient % dxclient -h
Usage: dxclient [options] [command]

DXClient is a CLI tool to manage tasks such as deploying portlets, script applicatic
s to target HCL DX deployments. This tool is capable of taking artifacts developed ]
pported on-premise platforms in standalone, cluster or farm topologies and Kubernete
perience 9.5 CF192 and higher only.

Common command arguments that are listed in `config.json` can be pre-configured via
hrough command line. If so, it will override the values in `config.json` and execute
  for Kubernetes environments is available under the 'samples/sample-configurations'

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

6. You may optionally try it now, using the version compare tool. 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 **dxclient version-compat -h**.

```
dxclient % dxclient version-compat -h
Usage: dxclient version-compat [options]

- Show version compatibility details between DX Core and DXClient.

Options:
  -hostname, --hostname <value>                Host name of the DX Core server (default: "")
  -dxConnectPort, --dxConnectPort <value>      Port number of the cw_profile, for any Kubernetes Environment default port is 443
                                               (default: "10202")
  -dxConnectProtocol, --dxConnectProtocol <value>  Protocol of the cw_profile (default: "https")
  -dxConnectUsername, --dxConnectUsername <value>  Username for the cw_profile (default: "")
  -dxConnectPassword, --dxConnectPassword <value>  Password for the cw_profile (default: "")
  -h, --help                                   display help for command
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 SoFy DX Solution Modules instance, under the **Sandbox Links** – **Configuration Wizard** (also used for DXConnect). You may find the host using any of the Open links. Both the dx and dxConnect parameters use this host, user ID and password.

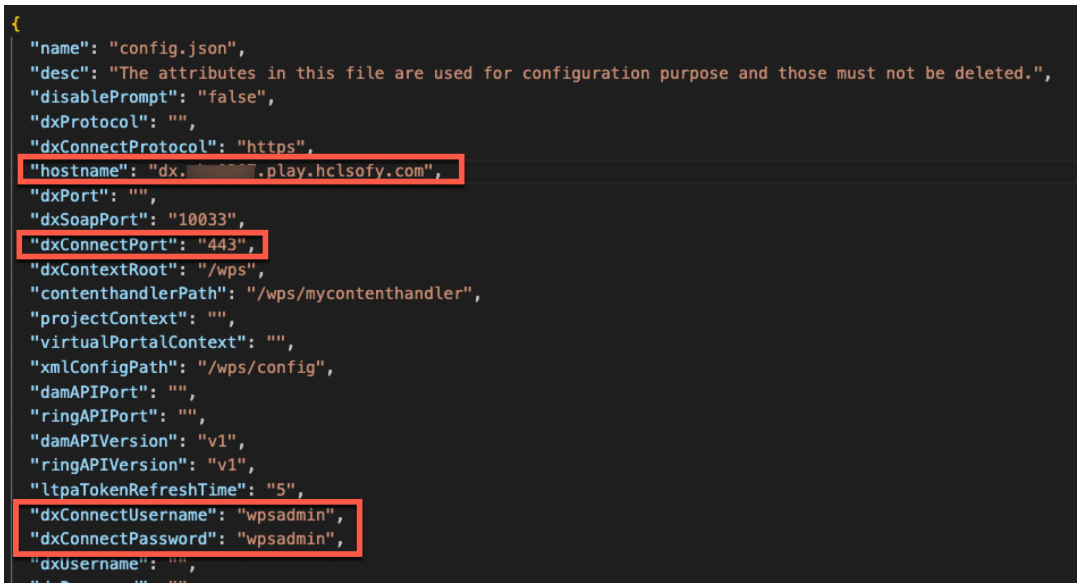


8. In this lab, you will connect it to your HCL SoFy instance. This is running on https, so port 443. Run the command file with these parameters and value, in this case (update with your host) **dxclient version-compat -hostname dx.sbx0000.play.hclsofy.com -dxConnectPort 443 -dxConnectUsername wpsadmin -dxConnectPassword wpsadmin**. Notice that in this example, the DX server was running CF220.

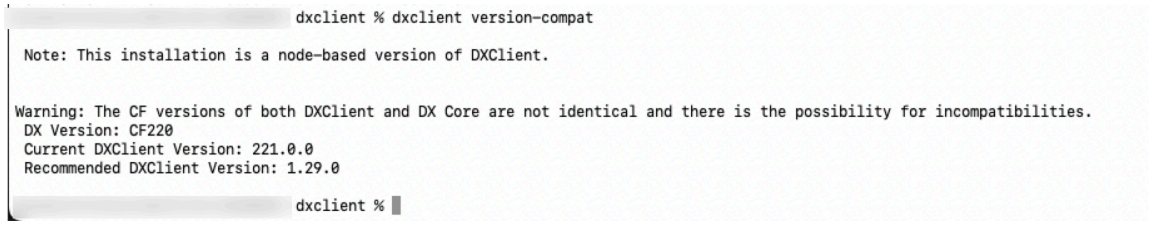
```
dxclient % dxclient version-compat -hostname dx.       .play.hclsofy.com -dxConnectPort 443 -dxConnectUserna
me wpsadmin -dxConnectPassword wpsadmin

  Note: This installation is a node-based version of DXClient.


Warning: The CF versions of both DXClient and DX Core are not identical and there is the possibility for incompatibilities.
  DX Version: CF220
  Current DXClient Version: 221.0.0
  Recommended DXClient Version: 1.29.0

dxclient %
```


9. 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.

10. To avoid having to add all these parameters to each command line, you may set them in the **store/config.json** file of your dxclient directory. Use your favorite text editor and apply this for example to the **hostname, dxConnectPort, dxConnectUsername** and **dxConnectPassword** entries. Then **Save** it:



11. Now you may run the Version Compatibility command without these globally set parameters, to check if your DX server is on the same level as your DXClient. 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.



12. You are ready to use it now! In other labs, you may learn how to apply it for specific resources that may require additional parameters.

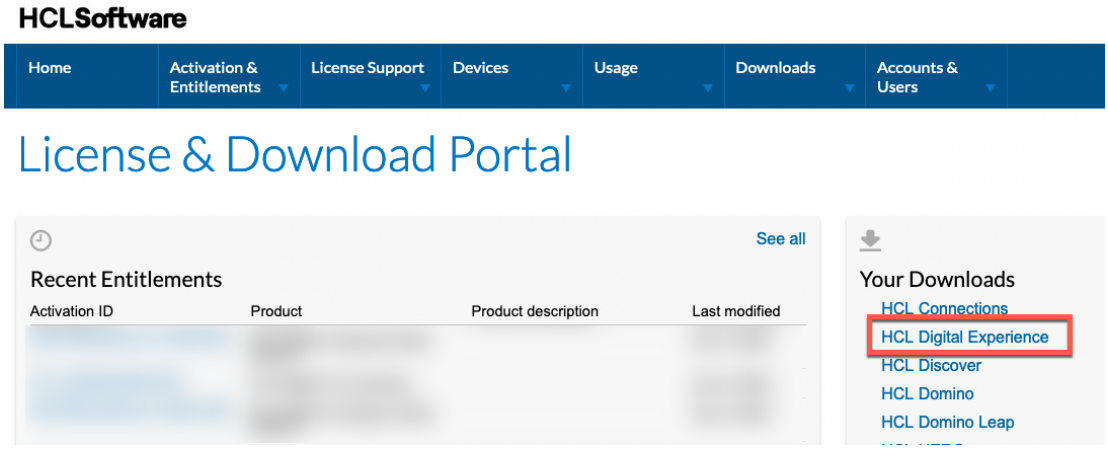
You have successfully configured and tested the DXClient !

Optional Part 3: Download, extract and deploy DXClient Container Package

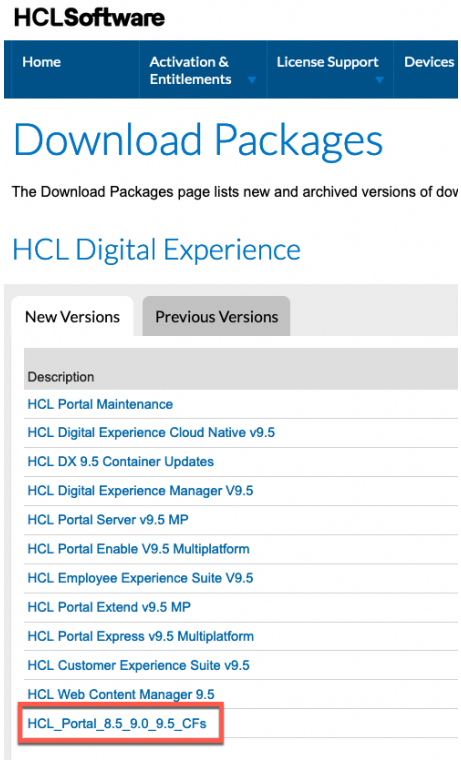
Use this optional part, if you prefer to use the DXClient container package. You will learn how to download and extract it, and then deploy it to your workstation. Optionally, you may learn how to further configure this to work in other places more easily, if you are working with different artifacts.

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.

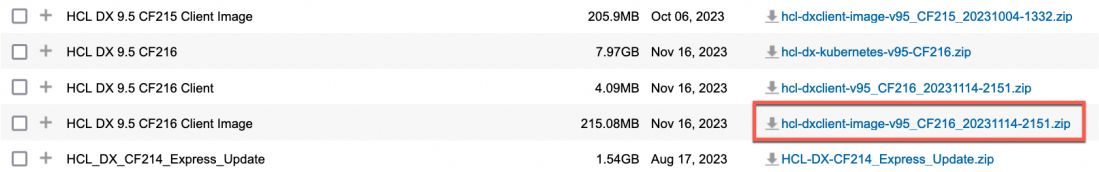
3. Log on to the License & Delivery Portal site: https://hclsw.co/flexnet. Under **Your Downloads**, scroll down and select **HCL Digital Experience**.



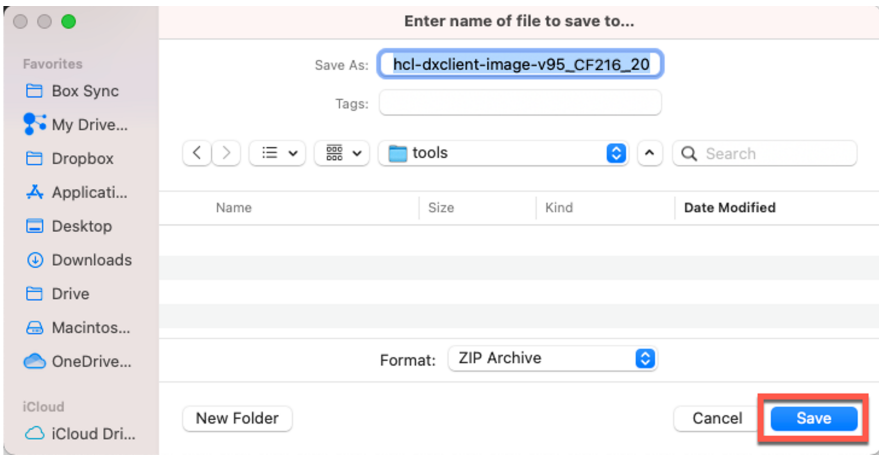
4. In the list search for **HCL_Portal_8.5_9.0_9.5_CFs** and click to access.



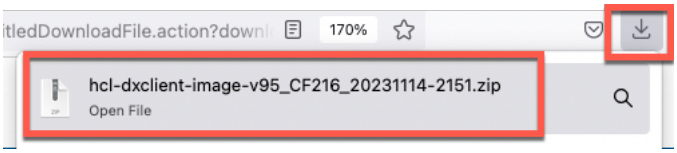
5. Then scroll and search for the version of the **HCL_DX_CFXXX Client Image** that matches your server and click on Client Image **hcl-dxclient-image-v95_CFXXX_XXXXXXXXX-XXXX.zip** to download it, in this case it is shown for CF216.



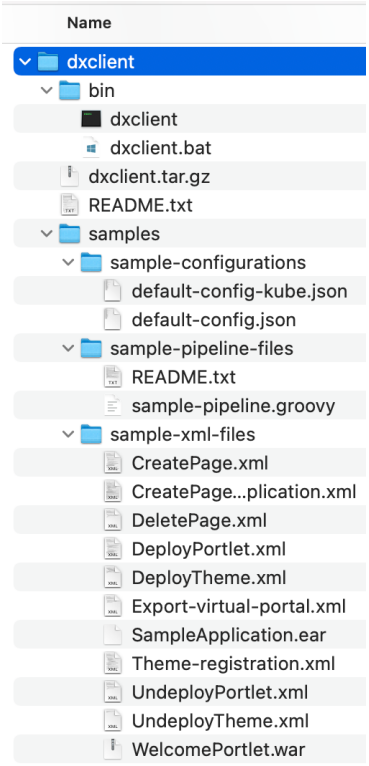
6. It asks you to select a place to save it. Select a working directory from which you wish to use the DXClient, for example a tools folder as shown here, and click **Save**.



7. 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.



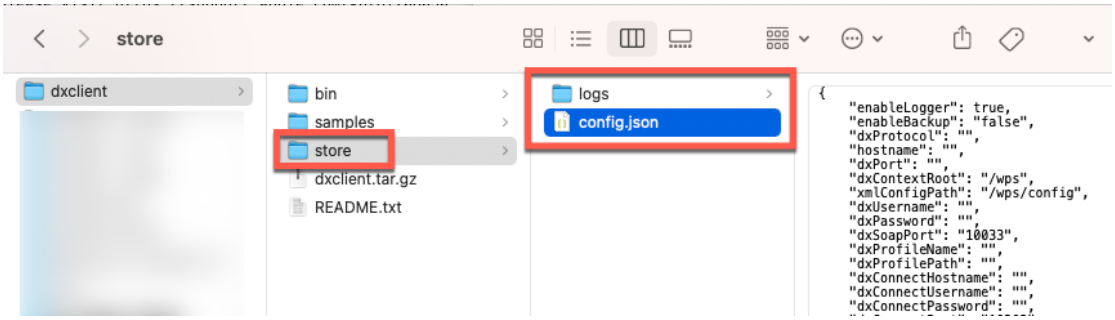
8. This should give you this folder structure with dxclient on the top. Have a look around the different files and folders. Under the **bin**, you will find the executable file for Linux/Mac and Windows. In the **samples** folder, you will several sample files to get started more easily, and the **README.txt**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



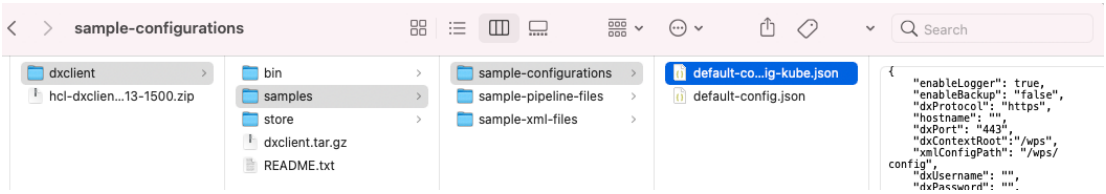
9. Then deploy it to your workstation as a Docker image. You need to have Docker installed and set up in your workstation. There are several places that gives you access to download and deploy Docker. For example from https://docs.docker.com/get-docker/. You may also use any container runtime that implements 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.
   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 %
```

13. To run the client, you need to use ./bin/dxclient command and it creates a folder structure like this with the bin, samples and your store working directory.



14. 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**.



15. When you run the dxclient -V command and you are using a Mac with an M1 processor, you may see this message.

```
herberthilhorst                    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
```


16. 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
```

10. 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. 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
```

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

```
dxclient — -bash — 103×41
Hilhorst-MacBook-Pro:dxclient hilhorst2$ export PATH=/Users/herberthilhorst/tools/dxclient/bin:$PATH
Hilhorst-MacBook-Pro:dxclient hilhorst2$
```

12. 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**.

13. 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 download, extracted, deployed the DXClient Image and configured your DXClient to easily work for different artifacts.

Conclusion

Using this lab tutorial, you have learned how to deploy the DXClient tool from an NpmJS registry , set the global configuration, optionally tested it to restart your server. You may also have learned how to deploy it as a Docker image and to make it easily work for different artifacts. You may use the Help Center 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 221 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.

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.

