Micro Focus Desktop Containers 20 w/ Application Streaming Add-on Evaluator’s Guide

Sept 2, 2020

Micro Focus Desktop Containers 20 w/ Application Streaming allows you to package applications into isolated containers to speed testing, deployment and updates of applications while also enabling those applications to be streamed to any device that runs the Turbo.net client or has an HTML5 browser.

How to Evaluate Micro Focus Desktop Containers

This section presents you with the high-level steps for evaluating Micro Focus Desktop Containers with the Application Streaming add-on. It links to other sections to help you find the step-by-step instructions required for a robust evaluation. The key steps covered by this guide are:

1. “Understand the resources required for the evaluation” on page 2
   The first task is to understand the hardware and software resources you’ll need to complete the evaluation successfully. So of these are mandatory, while others are optional depending on the capabilities you wish to evaluate.

2. “Install and configure Micro Focus Desktop Containers Server” on page 3
   Micro Focus Desktop Containers Server provides a web-based portal to the end user that allows them to launch applications that either run on the Windows PC or stream to a non-Windows or personally owned device. This step of the evaluation helps you understand how to deploy a Micro Focus Desktop Containers Server in your evaluation environment.

3. “Package and customize applications with Micro Focus Desktop Containers” on page 6
   Packaging applications as desktop containers allow you to more quickly test and deploy applications. It also helps you to eliminate conflicts between software and run older software that may conflict with the underlying Windows operating system. This section demonstrates several means of packaging applications with Micro Focus Desktop Containers.

4. “Publish and customize applications” on page 10
Once you’ve created a basic desktop container you can package the application as a standalone executable, portable executable or svm. You can also customize the application as needed and then publish the application to the software distribute mechanism of your choice. In this section you’ll learn how to build standalone executables that can be distributed with ZENworks or publish applications for delivery via the Micro Focus Desktop Containers Server.

5. “Execute the applications on Windows end-user devices” on page 14

Once you’ve packaged applications you are ready to switch hats and become the end-user. This section illustrates how end users can access desktop containers on Windows devices.

6. “Execute the applications on non-Windows, unmanaged or remote devices” on page 15

With the new Application Streaming add-on subscription for Micro Focus Desktop Containers 20 you can deliver desktop containers to your iOS, Android, Windows, MacOS, Linux, ChromeBook and other platforms that support an HTML5 browser. This section illustrates how to extend the reach of Desktop Containers to these devices.

7. “Use built-in reports and dashboards” on page 18

Once you’ve deployed applications via Micro Focus Desktop Containers Servers you can use the built in dashboards and reports to understand how applications are being used in your environment. This section looks at what’s available with the built-in reporting and dashboarding capabilities of the server.

Understand the resources required for the evaluation

Here’s a heads up on some of the resources you’ll need in order to run through this evaluation. All of the devices below must be connected to the same network and able to communicate with each other of IPv4. It is recommended that the server devices use a statically defined IP address and have a properly defined DNS environment. The following devices are needed for the evaluation:

- “Micro Focus Desktop Containers Server and Application Streaming Server” on page 2
- “Micro Focus Desktop Containers Packaging Machine” on page 3
- “Active Directory Domain Controller” on page 3
- “End-User Devices” on page 3

Micro Focus Desktop Containers Server and Application Streaming Server

The Micro Focus Desktop Containers Server provides the web interface necessary for users to run applications from anywhere. It also acts as the repository for all of the applications, making it possible to deploy those applications to users on demand. The Micro Focus Desktop Containers Application Server is a Windows Server running Remote Desktop Services that will be used when you want to stream applications to non-Windows devices or Windows devices that are not corporate owned or remote. In this evaluation you will use a single server to provide both of these roles as well as to act as the Database server. In a production environment you would typically have a dedicated database server, one or more portal servers, one or more hub servers and one or more application streaming servers to allow you to scale the system.

For the purpose of this evaluation you need a Windows 2016 or Windows 2019 server and Remote Desktop Services licenses. This server should have at least 16 GB of RAM and quad core processor. It is also recommended you install a modern web browser such as Chrome, Chromium Edge or Firefox on the machine.
Micro Focus Desktop Containers Packaging Machine

The packaging machine is where you will install the Turbo.net client and the Micro Focus Desktop Containers Studio application so that you can create your containerized applications. It is recommended that this machine be a virtual machine so that you can take a snapshot and easily roll-back to a clean state before each packaging session.

For the purpose of this evaluation you need a Windows 10 virtual machine that you can install the software on.

Active Directory Domain Controller

Micro Focus Desktop Containers supports both Active Directory and NetIQ eDirectory as user sources, enabling your users to use their corporate credentials to login to the Desktop Containers Portal.

For the purpose of this evaluation you need a separate Windows 2016 or Windows 2019 server acting as an Active Directory domain controller. The machine that the Micro Focus Desktop Containers Server is on should be a member of the domain. For information on setting up Active Directory, check out this video tutorial.

End-User Devices

The final core evaluation requirements are the end user devices that you want to execute desktop containerized application on or stream them to. This allows you to see the end user experience.

For the purpose of this evaluation we recommend that you have the following:

- A Windows 10 device (we'll use your packaging machine)
- A Mac OS device
- An iPad or iPad Pro
- An Android device (tablet preferred)

Install and configure Micro Focus Desktop Containers Server

To install and configure the Micro Focus Desktop Containers Server with the Application Streaming Add-on enabled, you need a Windows 2016 or Windows 2019 server configured. This server should be a member of your Active Directory domain. These instructions assume that you have completed the basic steps of bringing up a Windows server, assigning it a static IP address, joining the Active Directory domain and applying the latest Windows updates. Once you have completed these steps do the following:

1. Configure the server as a Windows Remote Desktop Services server. The Application Streaming Add-on for Micro Focus Desktop Containers allows you to deliver applications to an RDS farm and then stream applications to your non-Windows, remote or personally owned devices. To do this you must have a properly functioning Terminal Server. This step helps walk you through setting up the Remote Desktop
Services. In production, you would typically have one or more application servers that were separate from the Micro Focus Desktop Containers Server. To minimize the resource requirements of the evaluation, we run the hub, server and application streaming server components all on a single server.

2 Now that you have Remote Desktop Services properly configured, it’s time to install the Micro Focus Desktop Containers Server software on the machine. In production you may choose to have the server, hub, database and application streaming software on separate devices to ensure proper load balancing and fault tolerance. For simplicity we will use a single server in this evaluation.

   2a Login to your Windows 2019 server as a Domain Administrator.

   2b Copy the mfdc server installer to the Windows 2019 server.

   2c At the Welcome screen, click Next.

   2d At the License agreement screen, select the I have read and accept the terms and conditions checkbox, then click Next.

   2e At the License Certificate screen, click Next to accept the default evaluation certificate. This is good for 60 days.

   2f For the Destination Folder, accept the default by clicking Next.

   2g On the Server Roles screen, choose Standalone. In a production deployment you would typically install the Hub and Portal on one server and the Application role on other servers. But to keep things simple we’ll run them all on a single server for this evaluation.

   2h At the pop-up dialog, click Yes. This indicates that the Windows RDS role will be installed on the server as part of the install process.

   2i At the Web Service Configuration page, enter the full DNS name that end user devices will used to access the server. If you have a certificate (either issued by your AD controller or another CA) then choose https from the drop down and specify the path to your certificates. If you choose to use http you won’t be able to test the iOS launcher functionality. When you finish completing the form, click Next.

   2j At the Administration Account screen, make a note of the username you enter and then enter a password twice and make a note of it. This is used to authenticate to the admin console. Then click Install. This will install the Microsoft SQL server, MFDC server and RDS components and then reboot the machine.

3 Next let’s login to make sure the administrative interface is available. To do this browser the https://<your server>/admin. This should display a login prompt where you can login as the administrative user you specified during the server install.

   3a When the server reboots, log back in as a domain administrator.

   3b Open a browser and browse to https://<your server>/admin

   3c Enter the administrative credentials you noted during the installation of the server.

4 Configure the server to allow authentication by users in the Active Directory domain. To do this:

   4a In the admin console, go to the Users tab.

   4b Click Directory Services.

   4c Click the Add Service button.

   4d In the Name field, enter your domain name.

   4e In the Login Prefix field, enter the domain name.
4f Click the **Test Local** button. This uses the server’s domain credentials to read the users and groups from the domain. If the Succeeded message appears, scroll down and review the other settings and then click **Save**.

4g Click the ![refresh](https://example.com) button to initiate the synchronization of users from the domain. This may take a few minutes depending on the size of your domain. When the icon stops spinning click Users and Groups and verify that your domain users and groups are now listed.

5 Verify that you can authenticate to the end user portal, but browsing to https://<your server> and then login as one of your Active Directory users.

5a Open a new browser tab in Incognito / InPrivate mode.

5b Browse to **https://<your server>**

5c At the login prompt enter a domain user’s username and password, then click **Sign In**

5d You should see the basic portal shown below:

*Figure 1  Initial Portal Screen*

![Initial Portal Screen](https://example.com)

This completes the basic configuration of the Micro Focus Desktop Containers Server. Next let’s look at how to package applications as desktop containers.
Package and customize applications with Micro Focus Desktop Containers

There are multiple ways to create and customize applications with Micro Focus Desktop Containers. In this section of the evaluation we’ll look at a several of these methods to get you started. The tasks in this section include:

- “Install the Turbo.net Client on your Packaging Machine” on page 6
- “Install Micro Focus Desktop Containers Studio on your Packaging Machine” on page 6
- “Package an application through application monitoring in Studio” on page 7
- “Package an application using the Turbo.net hub in Studio” on page 8
- “Customize an application using the Run and Merge Capability” on page 9
- “Package an application with the Turbo.net command line” on page 10

In addition to the basic evaluation tasks presenting in this document, there are many advanced configuration capabilities that can be configured to determine how your applications function when containerized. For more information on these advanced capabilities refer to the Micro Focus Desktop Containers product documentation.

Install the Turbo.net Client on your Packaging Machine

In order to package applications and publish them to your server you’ll need to install the Turbo.net client and the Studio software on your packaging machine. If you have not already done so, start by building a vanilla Windows 10 virtual machine with 8 GB of memory and a modern browser. Once you have then complete the steps in this section and the next.

1. Login on your Windows 10 packaging machine.
2. In a browser, browse to https://<your server>
3. Login as your administrative user
4. At the top of the screen in the orange banner click the Download link to download the Turbo.net Client installer.
5. When the browser finishes downloading the installer, close the browser. Then go to the Downloads folder and run the installer.

The Turbo.net client is now installed and allows you to connect to either the Turbo.net hub or your local Micro Focus Desktop Containers server. Next you need to install the Studio application on your packaging machine.

Install Micro Focus Desktop Containers Studio on your Packaging Machine

The Micro Focus Desktop Containers Studio application is your primary tool for packaging applications into desktop containers. To install the Studio tool on your packaging machine, do the following:

1. Copy the installer to your packaging machine and run it.
2. At the Welcome screen, click Next.
3. At the End-User License Agreement screen, click Next.
4. At the Select Installation Folder screen, click Next.
5. Click Install.
At the UAC prompt (if you have UAC enabled), click Yes.

Click Finish to launch Studio. At this point if you have a valid license you can go to File > Options > License certificate... and paste in the contents of the license. If you do not then the embedded evaluation license is used.

Close Studio

Power off the virtual machine and then take a snapshot of the virtual machine so that you can revert to the clean state with turbo and the studio app after each packaging task.

You now have everything installed on your packaging machine to make it easy to package applications. The next several sections will walk you through the process of packaging applications as desktop containers.

Package an application through application monitoring in Studio

The first way you can package an application with Studio is by having Studio monitor the installation. It is also possible to do this by taking before and after snapshots with Studio. In this section you’ll see how easy it is to package an application. For this lab we’ll use the PuTTY SSH client for Windows. To do this:

1. Start your packaging VM and login.
2. Download the putty installer from https://the.earth.li/~sgtatham/putty/latest/w32/putty-0.74-installer.msi
3. Close the browser.
4. Launch Micro Focus Desktop Containers Studio.
5. Click Start Capture.
6. You are first asked for a location to store the output of the monitoring process. Create a new folder in the Documents folder called Putty.
7. Once you see the Recording system changes message, launch the installer for putty.
8. At the welcome screen, click Next.
9. At the Destination Folder screen, click Next.
10. At the Product Features screen, click Install.
11. If you have UAC enabled, click Yes.
12. Uncheck View README file, then click Finish. At this point you could launch Putty and make any additional customizations you wanted. Here we won’t because there are no customizations we need to make.
13. On the Recording system changes screen, click Stop.
14. When you receive the Select processes window, uncheck all process except msiexec.exe, then click OK.
15. Click Startup Files. This is the list of what happens when you run the desktop container. Notice that by default pagent.exe is run by the desktop container. We want to change that so both pagent and putty are executed. Find the putty line where the file putty.exe is listed and check the Auto Start checkbox.
16. Click on the website.url line, press tab and then press delete. We don’t need to be able to launch the website from the container.
17. Click on the putty.chm line, press tab and then press delete. We don’t need to be able to launch the help from the container.
18. Change the trigger for putty.exe from putty2 to putty. The triggers are command lines you can pass to the application to execute a particular application inside the container.
19 Click OK.
20 At the warning dialog, click Yes.
21 Change the Project type to Standalone/ISV Application so that it will build a simple windows EXE that you can run anywhere.
22 Click Build.
23 At the Choose the build output file, enter putty.exe, then click Save.
24 Click OK.
25 Browse to the executable and run it. Verify that both putty agent and putty are launched.
26 Close putty, then right click the putty agent icon in the system tray and choose Exit.
27 In Studio select File > Save to save the changes.

You have now successfully completed packaging and application with Studio. In it’s current form you could distribute this application to any Windows device simply by providing the user with the executable. This could be accomplished via removable media, a network share, or your normal software distribution solution.

Package an application using the Turbo.net hub in Studio

The next type of application we will package is an application from the Turbo.net hub. The Turbo.net hub is a repository of pre-package applications that you can use as a starting point for your own applications. In this section we will use Turbo.net to build the Filezilla application. To do so complete the following:

1 In Studio, select File > New.
2 From the Advanced tab, select Import Configuration.
3 Select Turbo.net and then click Next
4 If you do not yet have a Turbo.net account, go to https://turbo.net and create one.
5 Expand Tools > File Transfer > FileZilla
6 Select version 3.35.2, then click Next.
7 In the Destination folder field, browse to Documents and create a FileZilla folder and select it.
8 Click Next.
9 Studio will now use the Turbo.net client to pull the application to the local machine.
10 Click Finish.
11 On the Home tab, notice that an output file has been set.
12 Click Build to build the application.
13 Launch filezilla.exe and verify that FileZilla loads. Notice that you receive the welcome screen and likely an update notification.
14 Click Close on the update notification.
15 Close Filezilla.
16 In Studio, click File > Save to save the changes.

You now have your second desktop container. However, as you saw it checks for updates and may need other customizations such as adding a known FTP site to the site manager. In the next section you will do this.
Customize an application using the Run and Merge Capability

As you saw with the Turbo.net hub it is very easy to build a common application. However, you’ll often times want to customize that application. To allow for this Studio features the Run and Merge capability. In this section you will use Run and merge to add an FTP site and disable automatic checking in your Filezilla application. To do this follow these steps:

1. In Studio, select File > Log out of turbo.net
2. Select File > Login to MFDC Server
3. In the username field enter your administrative user, then enter the password. In the Server field enter <your server>
4. Click the Run and Merge button. This publishes the application to the local repository and then uses the Turbo.net client to execute the application.
5. Click OK at the welcome message and then Install Update at the update notification.
6. When Filezilla restarts
7. Select Edit > Settings.
8. Select Updates from the setting page.
9. Change the Check for FileZilla updates automatically dropdown value to Never.
10. Click OK. This will prevent the FileZilla app from checking for new updates.
11. Click File > Site Manager...
12. Click New Site.
13. Change the Site Name to Micro Focus FTP
14. Change host to ftp.novell.com
15. Click OK.
16. At the Remember passwords dialog, select Save passwords, then click OK.
18. You’ll see a container session is complete. Click Accept to create a new FileZilla xappl with your changes merged.
19. When prompted for a file name enter filezilla-eval.
20. Open the Documents\filezilla\files-filezilla-eval\layer1\@APPDATA@\FileZilla\filezilla.xml file.
21. Find the “Update Check New Version” setting and remove the entire setting from the file. This removes Filezilla’s knowledge of the update it previously discovered.
22. Save the file.
23. In Studio, make sure that Project Type is set to Standalone/ISV Application (EXE).
24. Click Build and Run
25. When prompted, set the file name to filezilla-nu.exe, then click Save.
26. The application is built and should run. This time you shouldn’t receive the prompt for the update. If you click the Site Manager icon in the tool bar you should see Micro Focus FTP in the drop down.
27. Close FileZilla.

You have now seen how to customize an application with Studio. Next we’ll look at how you can build applications with the Turbo.net command line.
Package an application with the Turbo.net command line

Another common way to build and customize applications is from the Turbo.net command line. In this section you will use the turbo.net hub to build and customize a Firefox 38.8 web browser. To do this complete the following steps:

1. Minimize Studio.
2. Run cmd.exe as Administrator.
3. Enter `turbo login <your turbo.net username>` and then enter your password.
4. Run Firefox ESR by typing `turbo run firefox-esr:38.8`. This pulls down the turbo.net Firefox image and then executes it in a container on your machine.
5. When Firefox loads, click the hamburger menu and click Options.
7. Click the Home button in the upper right to verify it goes where you expect.
8. Close Firefox. After a moment you’ll see Process exited with status 0.
9. Now you need to create an image from the container that has your changes in it. To do this type `turbo commit firefox-esr:38.8 eval/firefox-esr:38.8`.
10. Now you can export the application to an executable. To do this restore Studio.
12. Select Advanced > Import Configuration.
13. Select Local Repository, then click Next.
14. Select the `eval/firefox-esr:38.8` image and check the Import for standalone deployment checkbox.
15. Click Next.
16. For the Destination folder, create a `firefoxesr` folder under Documents, then click Next. This will import the image into Studio.
17. Click Finish.
18. Click Home.
19. Click Build and Run. This should build the executable and then start Firefox. Notice that it takes you to your Micro Focus Desktop Containers Server home page.

This completes our basic evaluation of building desktop containers. This is just a small sample of the power of Micro Focus Desktop Containers. For more information about packaging and customizing applications please refer to the Micro Focus Desktop Containers online documentation.

Publish and customize applications

Now that you have packaged your applications you are publish them so that end-users can access them. There are a number of methods for distributing applications that have been packaged with Micro Focus Desktop Containers. In this section of the evaluation we’ll explore the following:

- “Publishing applications to ZENworks from Studio (optional)” on page 11
- “Publishing applications to Micro Focus Desktop Containers Server from Studio” on page 11
- “Publishing applications to Micro Focus Desktop Containers Server from the Turbo.net Command line” on page 12
- “Customizing applications that have been published to Micro Focus Desktop Containers” on page 12
As with packaging, there are many different options that can be specified as part of your deployment of the applications. For a more exhaustive investigation of these, please refer to the Micro Focus Desktop Containers documentation.

**Publishing applications to ZENworks from Studio (optional)**

If you are a ZENworks customer and are using ZENworks to distribute software, you can use ZENworks to distribute desktop containerized applications. To make this easier the Studio application provides a built-in option allowing to easily publish applications as bundles. To do this:

1. In Studio, open the Putty `capture.xapp` file.
2. (Optional) If you want to restrict the devices where the app can run, to only those with the ZENworks agent registered in a specific zone, check the first two checkboxes and then specify the zone.
3. Select the Project type. If you select **Executable**, the bundle created will copy the EXE to the destination path you specify and create a Launch EXE action for when you launch the app. It will do this for each startup file that is part of the application. If you select MSI it will create an MSI that registers the icons, file extensions and verbs in the same way that doing a turboreg or Install to My PC does. In this evaluation, select **Executable**.
4. Select the zone and specify the server, admin username for the zone and password.
5. Specify the Bundle name for the application as **Putty-DCEVal**.
6. Specify the folder under the Bundles folder you want to create the bundle in. If you do not specify a location the bundle(s) will be created at the root of the Bundles folder.
7. (Optional) Specify the full name and path to a bundle group you want to add the bundles to.
8. Specify the destination path where the executable should be copied. In this case enter **c:\containers**.
9. Click **Publish Now** to publish the bundle to your ZENworks zone.
10. When you get a success message select **File > Save**. Then go to the ZENworks Control Center and notice that 4 bundles were created. Each will launch a different putty component.

This method of deploying local applications is useful for devices that are registered in the zone. The other method to is to deploy applications through the MFDC Server. The next two sections show you how to push applications to the server so that they can be distributed to your end user devices that are not managed by your endpoint management solution.

**Publishing applications to Micro Focus Desktop Containers Server from Studio**

For devices that are not managed by your endpoint management solution, personally owned devices, and non-Windows devices you’ll need to publish your applications via the MFDC Server. There are two ways to publish applications to the Server. The first is from Studio. To do this:

1. Click the **Publish to MFDC Server** button in Studio.
2. If you are not already logged in, you’ll be prompted for credentials. Make sure to specify administrative credentials for your server.
3. For the namespace, you can enter any value you want to use to help categorize the apps. In our case let’s just use `eval`.
4. For the Name, enter the name of the application, in this case `PuTTY`.
5. For the Release, enter the version number, in this case `0.74`.
6. Click **Next**.
When it finishes, click OK.

Click Next.

Click Finish.

Select File > Open and open the Filezilla eval xappl. Repeat these steps to publish Filezilla. For the version number specify 3.3.5. Use the same eval namespace and FileZilla as the app name.

You now have the applications available in the hub. They are not currently available to end users yet, unless they are familiar with the commandline syntax. In order for them to become available in the portal you’ll need to complete. “Customizing applications that have been published to Micro Focus Desktop Containers” on page 12

**Publishing applications to Micro Focus Desktop Containers Server from the Turbo.net Command line**

Another way to publish applications to the server, is to push them to the server from the command line. This can be useful in scripts as a way to automate the push of regular updates or as part of your devops process. Since we built Firefox from the command line, let’s push it to the server from the command line. To do this:

1. Run the command `turbo push eval/firefox-esr:38.8` from you administrative command line window.
2. When this completes the application will be available on the hub.

**Customizing applications that have been published to Micro Focus Desktop Containers**

You now have all of the applications available in the hub. Now you need to surface them to users via the Server Portal. To do that, do the following:

1. Open the server admin portal and login as your admin if you aren’t already.
2. Notice on the Hub screen that you now see the 3 applications you published to the server.
3. First, let’s change the default configuration so that if a Windows device is being used that it deploys the application locally instead of using the application server. Let’s also make sure that for non-PCs it’s configured to use Windowed mode by default for the optimal user experience.
   - 3a Click Domain.
   - 3b Click Settings.
   - 3c Change the Default Launch Mode for PC Clients dropdown to Run on My PC (Local)
   - 3d Change the Default Launch Mode for Non-PC Clients dropdown to Run in Cloud (Windowed)
   - 3e Click Save.
4. Next, as part of our Firefox application we want to have it launch with the Java Runtime Environment. Since this isn’t part of the base Firefox image we created, we need to have the JRE image available. To get that into the hub do the following:
   - 4a Click Hub.
   - 4b Click Add Repository > Import Repository.
   - 4c For the Repository Id enter oracle/jre
   - 4d For the Revision history length enter 5. This downloads the 5 latest images from the repository.
4e Click **Import**. It will take a few mins but will return you to the hub page and you should now see the JRE.

4f Click on the Java Runtime repository and notice the versions available. We want to use 8.211.

5 Now we need to publish the applications in the hub to a Workspace so that users can see them. To do this:

5a Click **Workspaces**.

5b Click **Manage** next to Default Workspace.

5c Click **Applications**.

5d Click **Add Application**.

5e Add all three applications.

5f Click **Close**.

6 Customize the PuTTY application so it shows the name PuTTY instead of PuTTY suite.

6a Click **Putty suite**.

6b Change the display name to **PuTTY**

6c Click **Save**.

6d Click **Cancel**.

7 Customize the Firefox application so that it includes Java, can only visit DNS names that are in the microfocus.com domain, and ensures that user customizations follow the user.

7a Click **Firefox**.

7b Click **Components**.

7c In the Component name enter **oracle/jre:8.211**, then click **Add**.

7d Click **Save**.

7e Click **Networking**.

7f Change Network Routes to **Disallow all sites except**.

7g In the table add a route for ***.microfocus.com**, then click **Add**.

7h Click **Save**.

7i Click **Storage**.

7j Turn on the **Sessions are persistent** slider. Also notice that you can use Storage visibility to prevent users from seeing unwanted drives. This is typically used in conjunction with the T: drive capabilities. For more information about T: drive please refer to the documentation as it requires a corporate Dropbox or OneDrive account we will not explore this in the evaluation.

7k Click **Save**.

7l Click **Cancel**.

Because we placed the applications in the default workspace they are available to all users in the workspace. This completes the customization of the workspace and applications that we explore here. Please refer to the online documentation for more information about other customizations such as licensing, advanced networking and isolation that can be configured.
Execute the applications on Windows end-user devices

After you have created your containerized applications and published them to an executable or to the Server, you are ready to test your applications. Micro Focus Desktop Containers can be executed on any Windows device in a variety of ways. The tasks in this portion of the evaluation include:

- “Launching applications from the Micro Focus Desktop Containers Portal” on page 14
- “Launching applications using the Turbo.net command line” on page 14
- “Registering applications with the local machine and launching registered applications” on page 14

Launching applications from the Micro Focus Desktop Containers Portal

Now let’s look at how to launch applications that you’ve published to the Desktop Containers Server. To begin with you’ll need to install the Turbo.net client and browser extension if they aren’t already deployed. The online documentation provides information on the command line switches to perform a silent installation of the Turbo.net client through a software distribution tool like ZENworks. It also provides information about how to push the extension to both Chrome and Chromium-based Edge using a group policy. In this evaluation we’ll just use your packaging machine as the Windows device where we launch apps.

1. Install the Turbo.net browser extension. To do this go to the Chrome extension store, find the turbo.net extension and add it. This can be done on either Chrome or Edge. For Firefox go to the Firefox extension library and add it.
2. Close any browsers you have running.
3. Open a new browser and browse to https://<your server>
4. Login as a domain user. You should see all of the applications that you published to the workspace.
5. Click on one and it should cache the container image to the device and then launch the application. Repeat this for each application
6. Notice that when you exit Firefox it synchronizes the session back to the server so that when you go to another machine it maintains the user settings.

Launching applications using the Turbo.net command line

You can also launch applications from the command line. This is useful if you wanted to create a bundle that launched with advanced command line switches or you were trying to troubleshoot the launch of the app. To launch an app from the command line use the command `turbo run <image name>:<version>` to launch firefox or any of the other applications, substituting the image name. To determine the image names and versions available you can use the following commands:

- `turbo search "` to list all of the images available on your hub
- `turbo releases <image name>` to see all of the versions of a particular application that is available.

Registering applications with the local machine and launching registered applications

Users can also chose to have containers installed on their PC. This registers any desktop or start menu shortcuts, file extensions and verbs that the application contains. It also lists the application in the Add/Remove programs control panel. The advantage of this method of installing the application is that the user can interact with the applications in the same way they would a local, non-containerized application. If you want
to force register an application for a user you can use your endpoint management software to execute the turboreg.exe command to register the application. See the online documentation for more about this option. If an end user wishes to register an application, this is how they do it:

1. In the server portal, right click Firefox.
2. Select **Install to my PC**.
3. When it finishes, click on the Start menu and notice that there is a Firefox icon.
4. Click the icon and notice that the containerized application is launched.
5. Close Firefox.

Because the application is also registered with HTML extensions if you right click an HTML file you’ll also have the option to open the file with the containerized application, as well as any locally installed browsers.

### Execute the applications on non-Windows, unmanaged or remote devices

In today’s world, users are working from more places, on more devices than ever. Being able to run their business applications on their corporate owned device while in the office just isn’t enough. With the Application Streaming Add-on subscription you can extend the capabilities of Desktop Containers to almost any device imaginable. This section examines how you execute applications across a wide variety of platforms.

- “Launching applications on un-managed Windows devices” on page 15
- “Launching applications on MacOS devices” on page 16
- “Launching applications on iOS devices” on page 16
- “Launching applications on Android devices” on page 16
- “Launching applications on other devices via HTML5 streaming” on page 17

### Launching applications on un-managed Windows devices

If you have users that are working or learning from home or on their own hardware, you may not want to have the application or the data on that PC. In that case you may want to consider delivering the application using Application Streaming. With application streaming the application is delivered via the Turbo.net client to a Windows terminal server and then the application UI is streamed to the user over the RDP or H.264 protocol. This capability works equally well on Windows, Mac, Linux, iOS, Android, Chromebook and more. To stream applications to Windows devices do the following:

1. Ensure that the Turbo.net client is installed on the machine as well as the Turbo.net browser extension.
2. Login to the server portal.
3. Right click an application and select **Run in Cloud (Windowed)**.
4. When you receive the prompt asking if the browser can open the extension, click the checkbox to always allow and then click Allow.
5. You will see the Turbo.net client connecting. The first time you launch the application is may take a few minutes as the application will be distributed to the terminal server and then executed. If you want to reduce this time there is an option at the workspace to pre-cache the application to the application servers.
The application should then appear. While it looks like it is running on your PC, it is actually running on the RDP server.

Close the application

Launching applications on MacOS devices

If you have MacOS devices that users need to access their applications from, you can follow much the same process.

1. Open your server portal in Safari or Chrome.
2. Login as the user.
3. Download and install the Turbo.net client.
4. Click the application you want to run. Similarly to Windows you may get a prompt about the extension calling an external program. Allow that.
5. The application should launch and appear to be running on the device, even though it is really running on the remote application server. Depending on the storage visibility configuration you can even see your local files and interact with them in the application as well as the T: drive.

Launching applications on iOS devices

The best way to launch a desktop container from an iOS device is to install the Turbo.net application from the app store. This can be either manually installed or you can use an MDM solution like ZENworks to deliver the application and configuration to the device. Once the application is installed, all you have to do is:

1. Run the Turbo.net app.
2. Login as a domain user, making sure to specify your server instead of turbo.net. In order for the iOS app to function properly you must be using HTTPS on the server.
3. Tap on the application you want to run.
4. The application will launch and you’ll see a small tutorial on how to use the application.

Launching applications on Android devices

The best way to launch a desktop container from an iOS device is to install the Turbo.net application from the play store. This can be either manually installed or you can use an MDM solution like ZENworks to deliver the application and configuration to the device. Once the application is installed, all you have to do is:

1. Run the Turbo.net app.
2. Login as a domain user, making sure to specify your server instead of turbo.net.
3. Tap on the application you want to run.
4. The application will launch and you’ll see a small tutorial on how to use the application. There is also a small toolbar at the bottom of the app as shown below:
Launching applications on other devices via HTML5 streaming

For other types of devices that do not have a Turbo.net client, such as Linux or Chromebook, or if your users don’t want to install the Turbo.net client; you can use the HTML5 streaming. This simply requires an HTML5 browser. To use HTML5 streaming do the following:

1. On your packaging machine (or whatever other machine you want to test from) login to the serve portal.
2. Right click the application and select **Run in Cloud (HTML5)**. This will start the application in the browser tab itself shown below:
3 Notice the toolbar in the lower right hand corner. This allows you to transfer files between your machine and the application server, print files to PDF, manage the clipboard and do tab switching as needed since these capabilities are not natively available in the HTML5 mode.

4 Close the application.

**Use built-in reports and dashboards**

After your users begin using the Micro Focus Desktop Containers Server and/or portable applications you’ll be able to track the use of applications across the domain using the in-built dashboards and reports. This section takes you on a tour of these capabilities.

1 Logout as the domain user from the server portal.

2 Login to https://your server/admin as your administrative user.

3 First review the server load dashboard by going to **Domain > Dashboard**. This shows the CPU, memory and session count for each of the servers in the MFDC Domain as shown below:
Next let’s look at the Workspace dashboard.

4a Click on **Workspaces**.

4b Click on **Manage**.

4c Review the workspace dashboard to determine your most launched applications and the usage summary as shown below:
5 Review the Reports available.
   5a Close the workspace admin page.
   5b On the main admin page, click Reports.
   5c Click on each of the reports to review what’s available.

In addition to the basic reports and dashboards, a domain is or will soon be available for ZENworks Reporting to allow you to create your own custom reports and dashboards. Watch for it on the MFDC and ZENworks Reporting communities pages.

Summary

This evaluation guide was intended to introduce you to the basic capabilities of Micro Focus Desktop Containers with the Application Streaming add-on subscription. Now that you’ve gotten a taste of what it can do, we highly encourage you to look at the applications in your environment that would benefit from containerization and the ability to stream them to any device your users are using. If you require additional assistance in completing your evaluation please drop us a line at zen@microfocus.com.

Legal Notice

For information about legal notices, trademarks, disclaimers, warranties, export and other use restrictions, U.S. Government rights, patent policy, and FIPS compliance, see https://www.microfocus.com/about/legal/.

© Copyright 2020 Micro Focus or one of its affiliates.