
FlexApp™ Click-to-Layer for Published Apps

Summary

ProfileUnity's FlexApp Technology includes exclusive Click-to-Layer capabilities and Citrix XenApp/Microsoft RDSH session isolation for FlexApp application packages. FlexApp's Click-to-Layer feature allows instant delivery of FlexApp layered applications on-demand upon the application open action by a user. The user simply clicks on the application's native icon on the Desktop or in the Start Menu and the application is instantly delivered via FlexApp and is ready for use. The option improves desktop readiness times compared to other layer delivery options because applications are not mounted unless they are in use. Now even dozens of layered applications can now be readily available in a user's Windows environment without needing to actually layer every available application that may not be used during a user's session.

FlexApp's Session Isolation feature provides enhanced delivery support providing session-based layering for published applications running on Citrix XenApp, Microsoft RDSH, and Microsoft RemoteApp. Before, all applications published on a server were visible to all connected users whether they need them or not. Now when FlexApp applications are delivered to a user's published desktop, other users are not automatically exposed to those applications. Every user's desktop can remain unique, even for users on the same Windows Server environment.

Click-to-Layer in a server environment offers an additional advantage. The first user to request an application to open will experience a slight delay as the FlexApp package is layered into the environment. However, once the application is layered into the server environment, the process does not need to be repeated. Subsequent users will not experience any delays on an application open.

This document explains how to configure FlexApp's Click-to-Layer feature for published applications.

Updating FlexApp Settings for Click-to-Layer with Published Applications

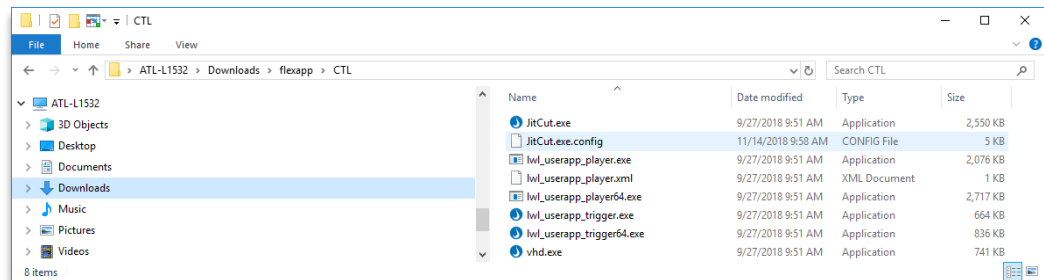
Before configuring Click-to-Layer with your preferred method of publishing applications, you will need to complete the following:

1. Package your application with the FlexApp Packaging Console v6.7 or later. Note: If the desired application has been packaged prior to FlexApp Packaging Console v6.7, the package can be updated for Click-to-Layer compatibility by opening the FlexApp Packaging Console v6.7, selecting **Edit the package's metadata** for the desired package and then clicking **OK**.
2. If not already done, click **Download or Deploy Client Tools** found under **Administration Settings > ProfileUnity Tools** of the ProfileUnity Management Console to your **Deployment Path** which is normally your \\[domain]\NETLOGON\ProfileUnity folder.
3. Enabling Click-to-Layer for published applications requires you to edit FlexApp configuration settings inside flexapp.zip that was just copied to the **Deployment Path** as part of the Client Tools. The default location for **flexapp.zip** is in the **NETLOGON\ProfileUnity** folder. Copy flexapp.zip locally and complete the following steps:
 - a. Unzip **flexapp.zip**.

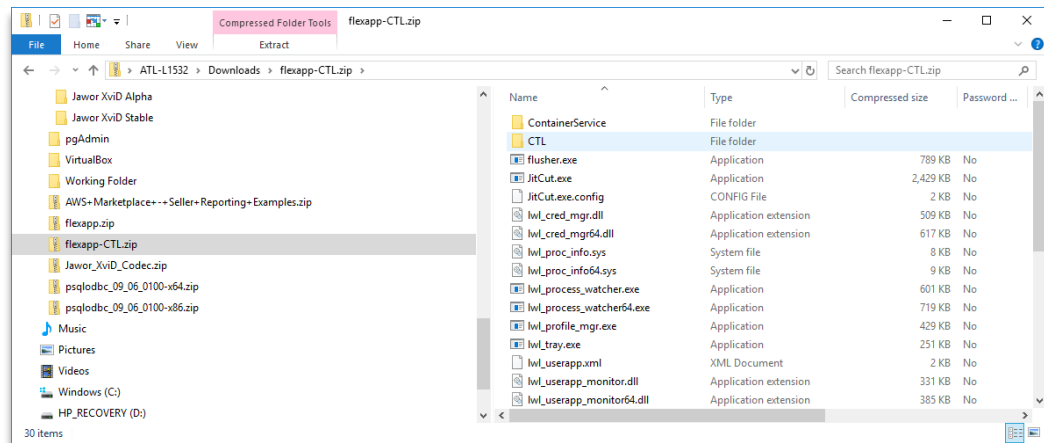
- b. Create a new subfolder under the unzipped FlexApp folder for the Click-to-Layer files called "CTL".
- c. Copy the following files from the FlexApp folder into FlexApp\CTL:
 - i. JitCut.exe
 - ii. JitCut.exe.config
 - iii. lwl_userapp_player.exe
 - iv. lwl_userapp_player.xml
 - v. lwl_userapp_player64.exe
 - vi. lwl_userapp_trigger.exe
 - vii. lwl_userapp_trigger64.exe
 - viii. vhd.exe
- d. Edit FlexApp\CTL\JitCut.exe.config updating JitCut.exe.config package validation settings by opening and changing the following setting from **True** to **False**:

```
<setting name="Validate" serializeAs="String">  
<value>False</value>
```

- e. Then replace **JitCut.exe.config** in the FlexApp\CTL folder with the modified version for Click-To-Layer.



- f. Rezip the contents of the FlexApp folder with the added \CTL subdirectory as flexapp.zip. You may rename this file to avoid overwriting the original flexapp.zip, but remember to copy it as “flexapp.zip” in the next step.

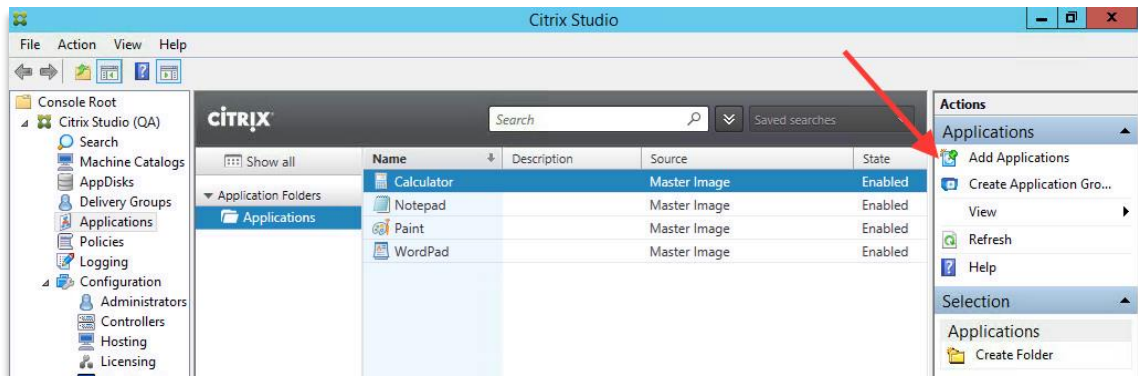


4. Go back to your environment’s **Deployment Path** (by default, this is \\[domain]\NETLOGON\ProfileUnity). Copy the modified flexapp.zip to your environment’s **Deployment Path**.
5. If the client software has previously been deployed to the end user machines in your environment, you will need to uninstall the existing client software. If the client software has not been previously deployed to the end user machines in your environment, the client software will be installed through a Group Policy computer startup script from the **Deployment Path** with the modifications that were made to flexapp.zip for FlexApp Click-To-Layer.

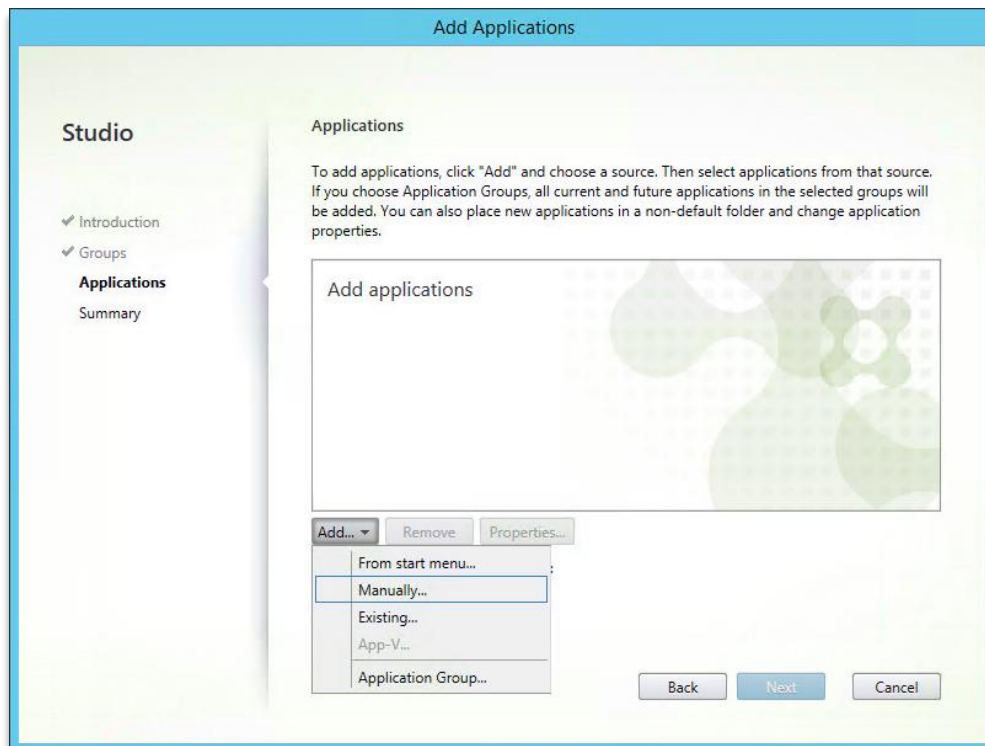
Now you are ready to continue the configuration process based on your choice of application-publishing brokers.

Citrix XenApp

1. Login to Citrix Studio and navigate to **Applications**.
2. Under Actions, click **Add Applications**.

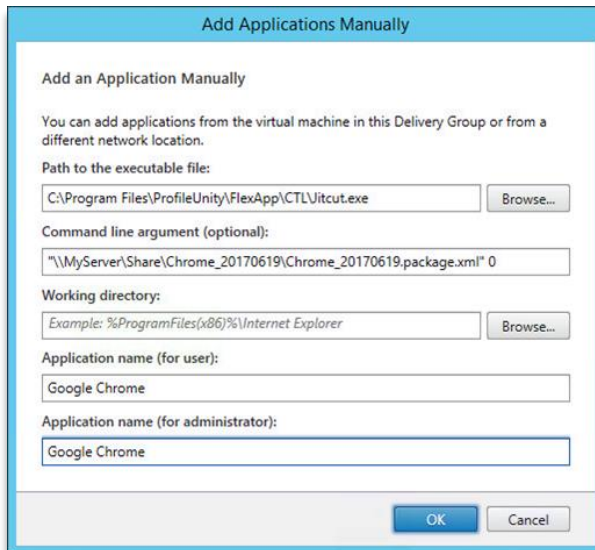


3. Proceed through wizard until getting to the Applications page.
4. Click the **Add** dropdown, and select the **Manual** option.

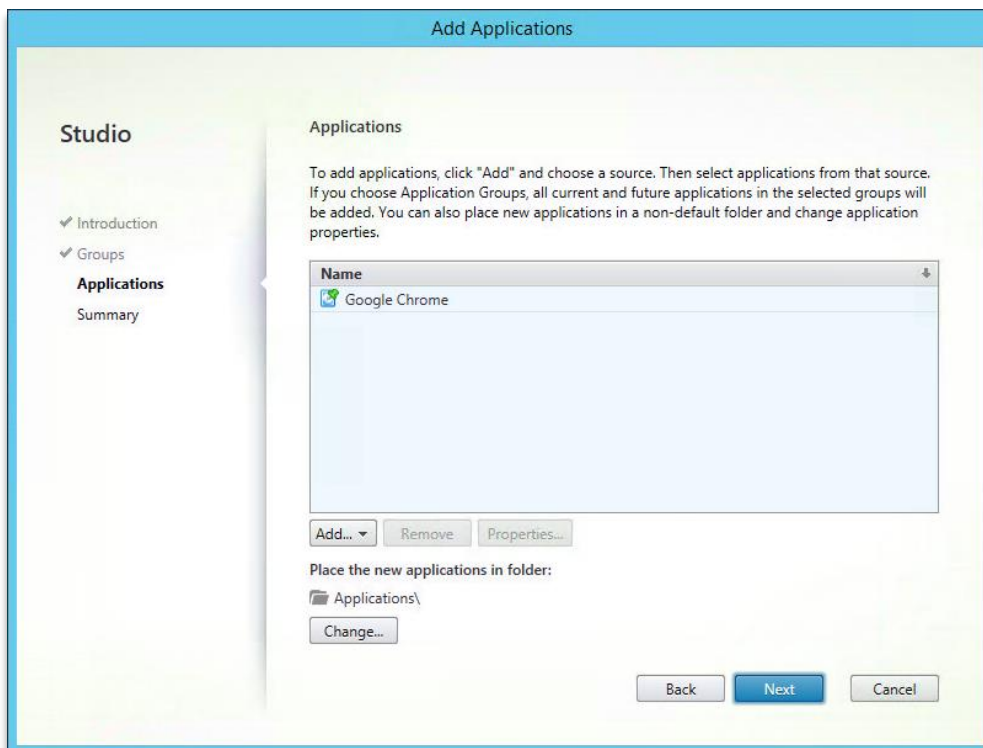


- For **Path to the executable file**, enter the full client-side path to the modified Click-to-Layer version of **JitCut.exe**:

C:\Program Files\ProfileUnity\FlexApp\CTL\JitCut.exe

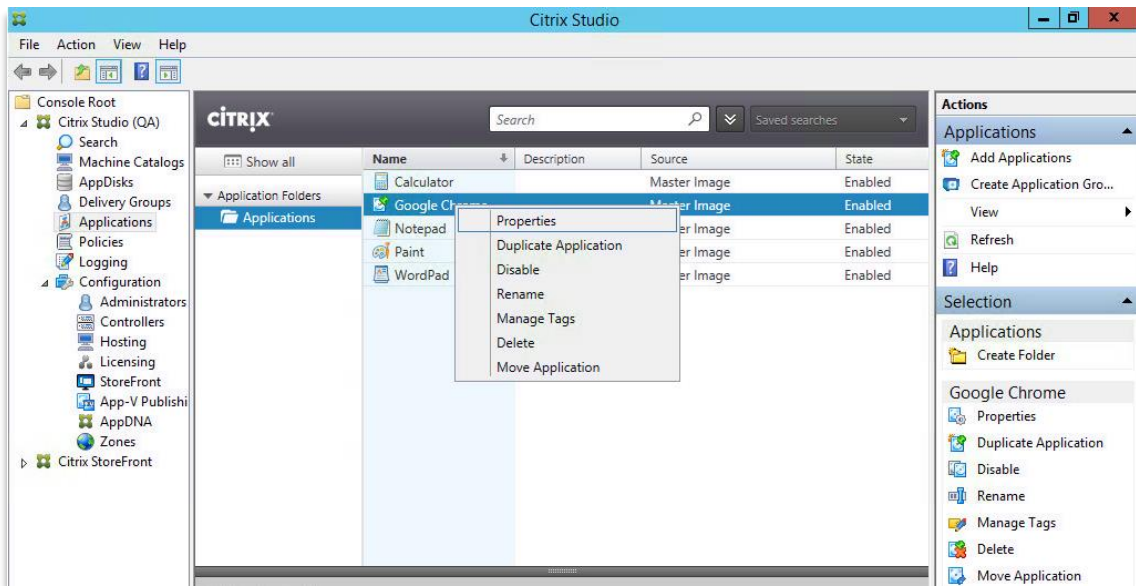


- In the **Command line argument** field, specify the path to the package xml (stored alongside the package FlexApp Packaging Console creates) using the following format:
"`\\MyServer\Share\Chrome_20170619\Chrome_20170619.package.xml`" 0
- Specify the desired Application names then press **OK**, returning you to the Add Applications main wizard.

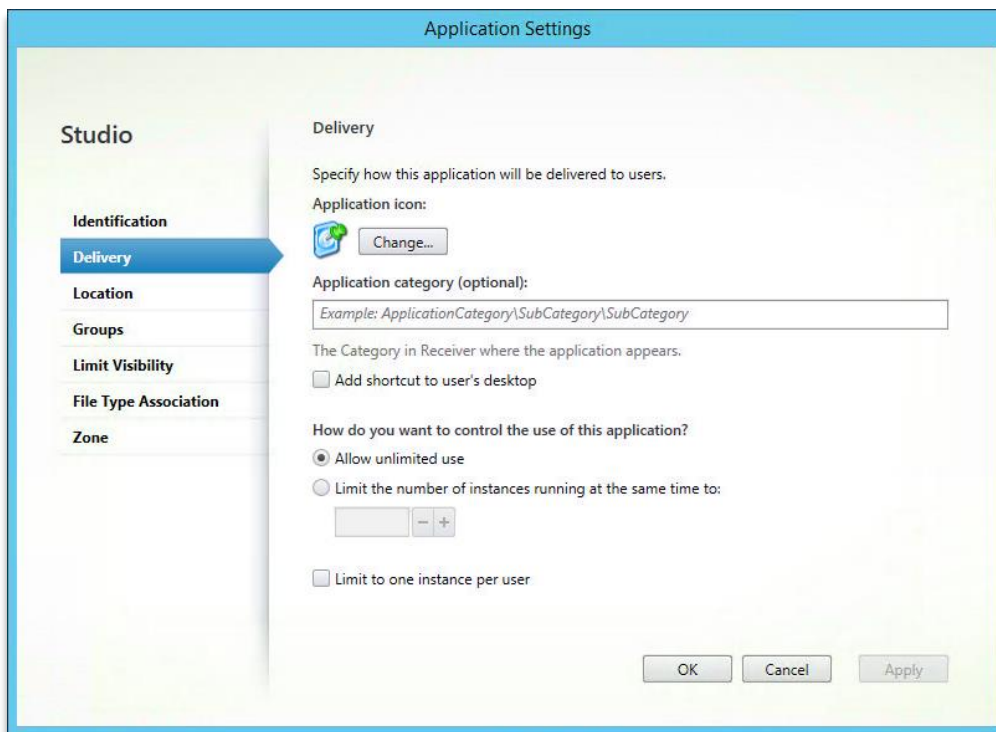


- Complete the Add application wizard and verify the application has been added successfully.

9. Right-click **Application**, and go to **Properties**.

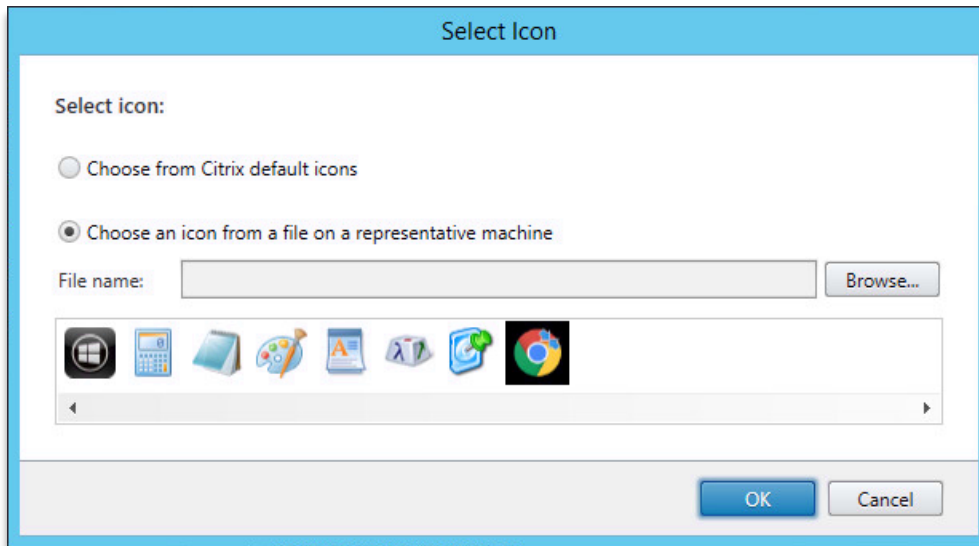


10. Select **Delivery** from the side menu.

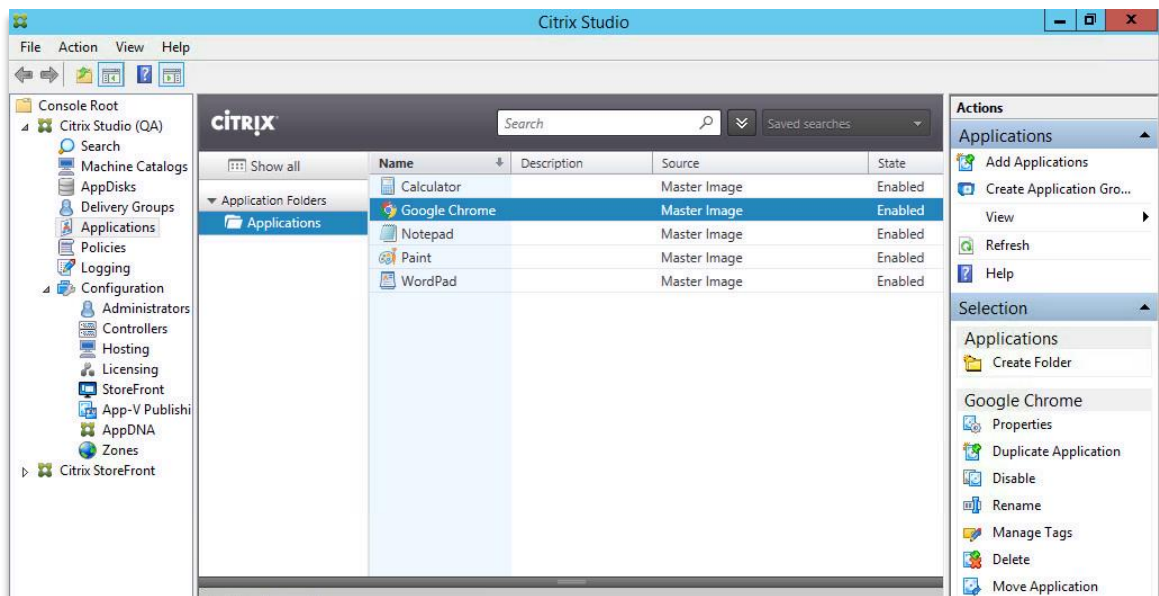


11. Under Application icon, click **Change**.

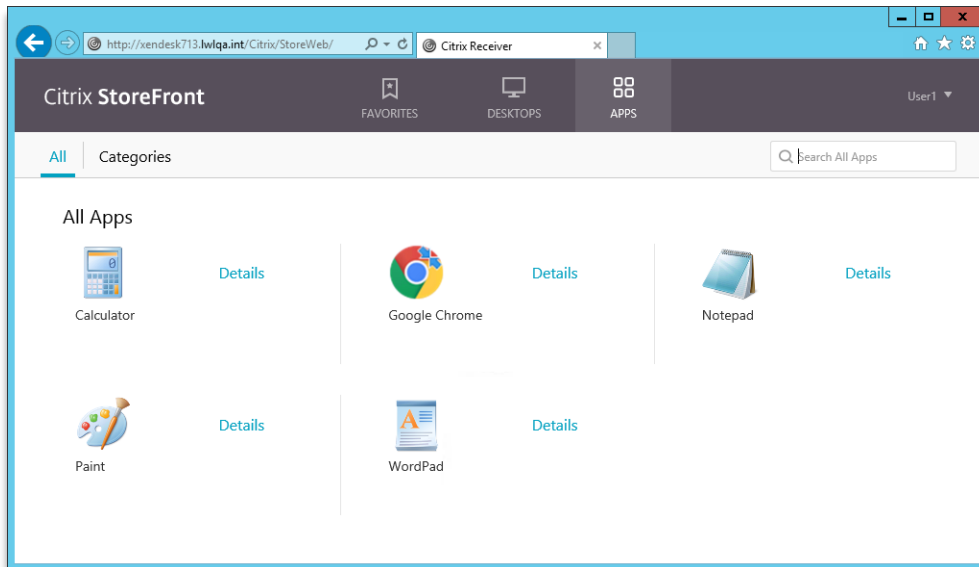
12. Select **Choose an icon from a file on a representative machine** and select the .ico file located in the package path (specified in step 9).



13. Press **OK** to save changes and verify the new icon is present.

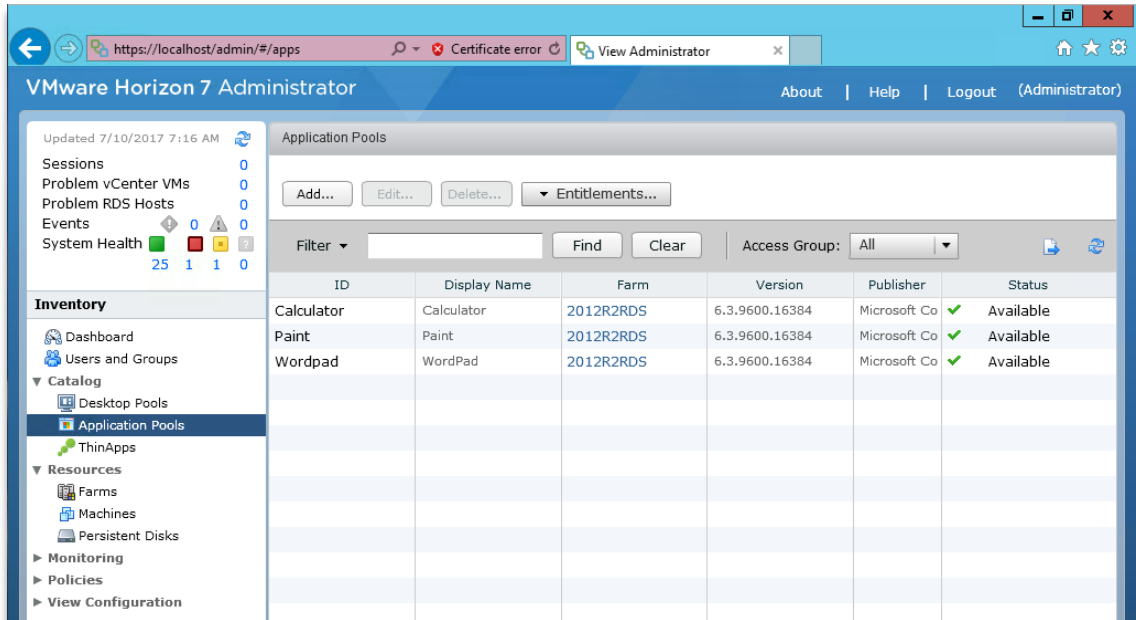


14. Configuration is complete. The application can now be assigned and launched from the Citrix StoreFront, the same as any other published application.

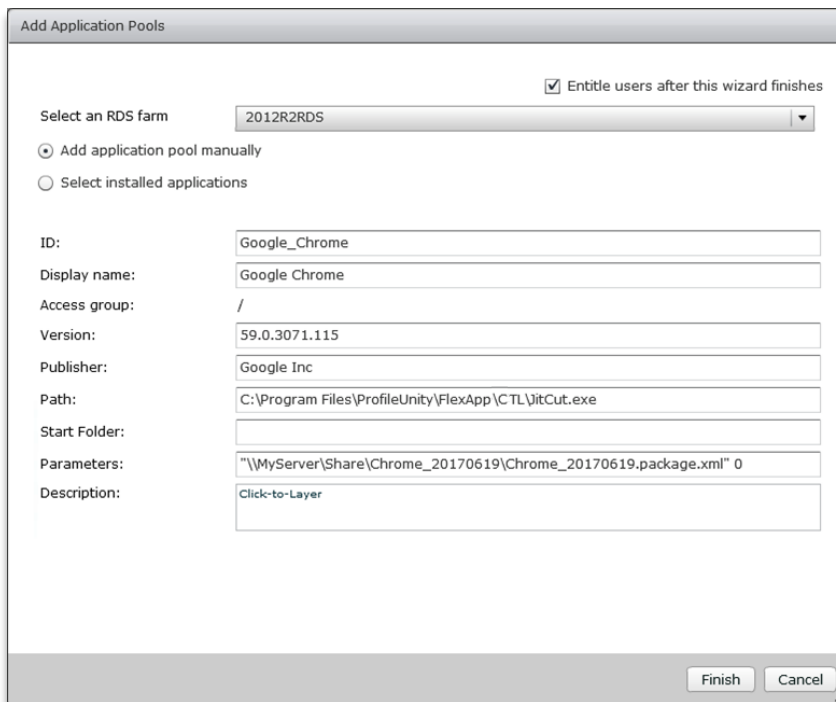


VMware View RDSH

1. Login to the VMware View Admin Console and navigate to **Application Pools**.

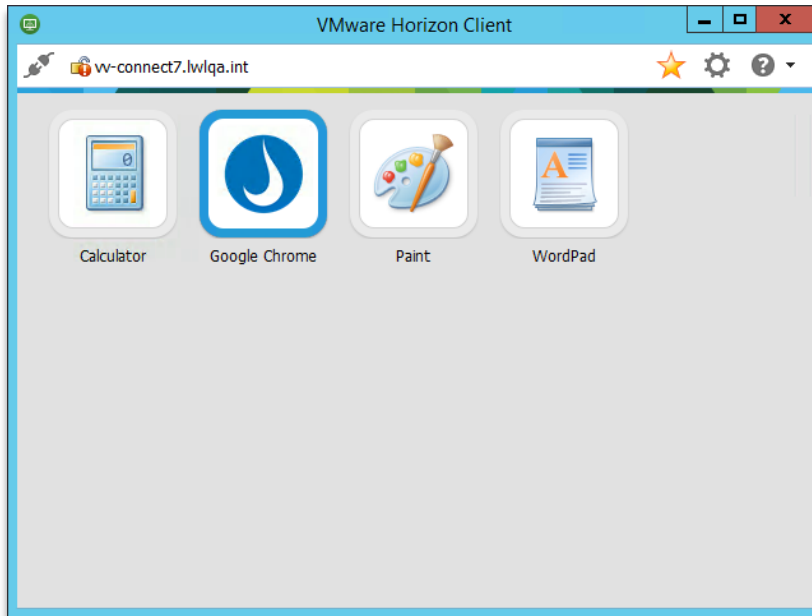


2. Click the **Add...** button to add a new application.
3. Select **Add application pool manually**.



4. Entire desired **ID**, **Display name**, **Version** and **Publisher**.
5. For **Path**, enter the full client-side path to the modified Click-to-Layer version of **JitCut.exe**:
`C:\Program Files\ProfileUnity\FlexApp\CTL\JitCut.exe`

6. In the **Parameters** field, specify the path to the package .xml (stored alongside the package FlexApp Packaging Console creates) using the following format:
`"\\MyServer\Share\Chrome_20170619\Chrome_20170619.package.xml" 0`
7. Press **Finish** to complete the application configuration process.
8. Configuration is complete. The application can now be assigned and launched from the VMware View Client, the same as any other application.

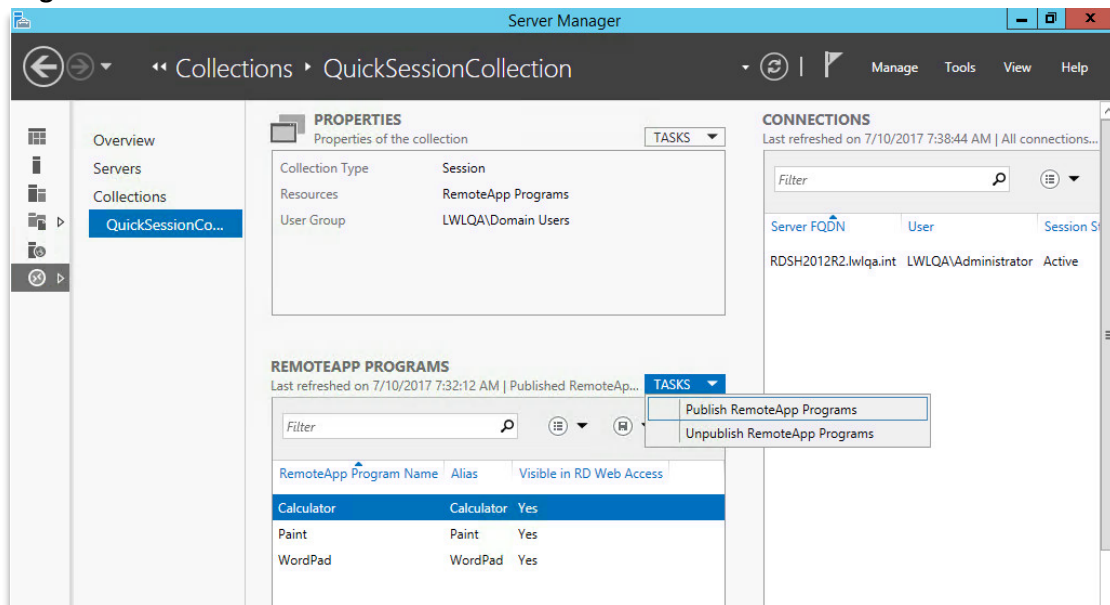


Note: Starting with VMware View 7.1, VMware has added experimental support for changing the default icons for published applications. Both VMware View 7.1 and PowerCLI 6.5 R2 are required. Once installed the icon can be changed by opening a PowerShell cmd prompt and performing the following:

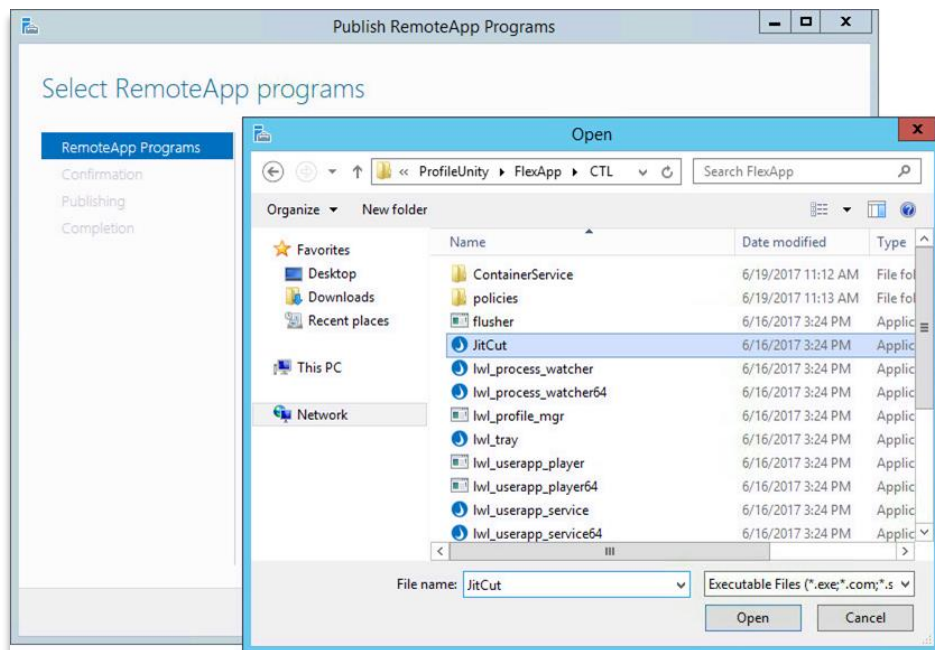
1. First load the required PowerCLI modules:
 - a. `Get-Module - ListAvailable VMware.VimAutomation.HorizonView | Import-Module`
 - b. `Get-Module - ListAvailable VMware.Hv.Helper | Import-Module`
2. Then connect to the VMware View connection server:
`Connect-HVServer -Server your.connectionserver.com`
3. Finally, set the Application icon to an image file (.ico, .png, .jpg, etc.):
`Set-HVApplicationIcon -ApplicationName Chrome -IconPath C:\MyIcons\chrome.ico`

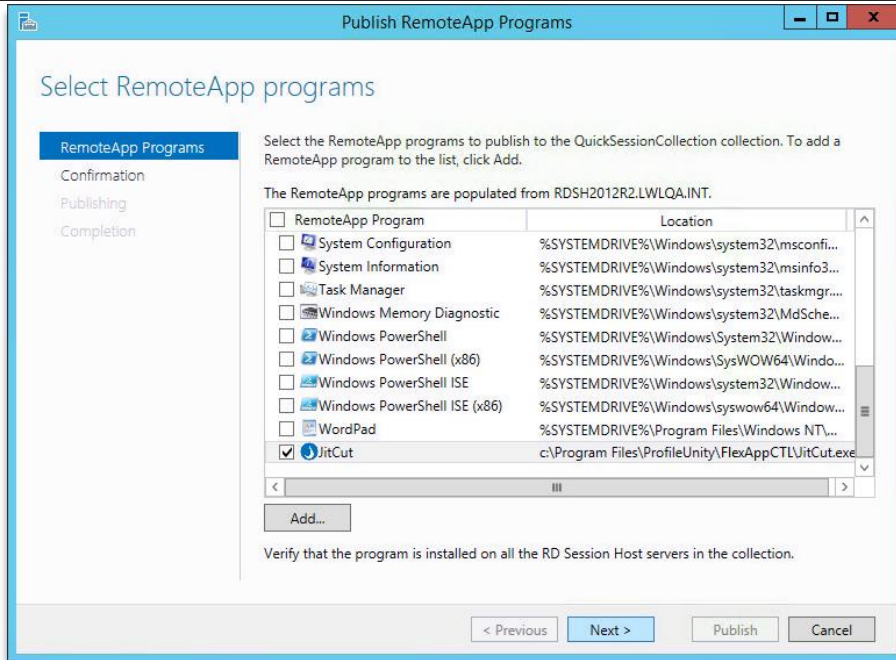
Microsoft RemoteApp

1. Login to the RemoteApp server and navigate to **Collections**.
2. Under **RemoteApp Programs** click the **Tasks** dropdown and select **Publish RemoteApp Programs**.

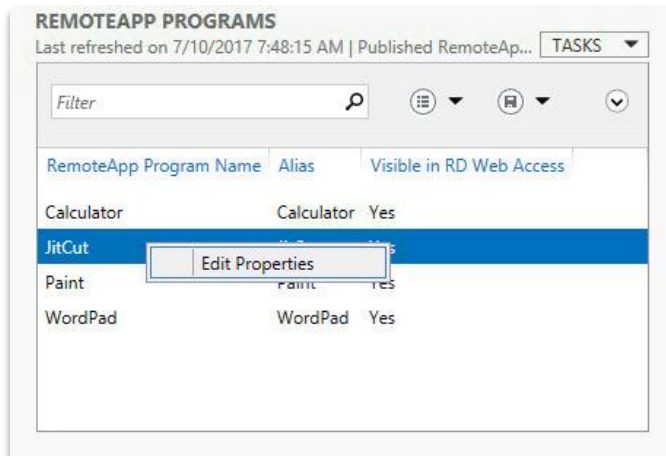


3. The Publish RemoteApp Programs wizard is launched. Click the **Add...** button to manually add an application.
4. Navigate to and select the modified Click-to-Layer **JitCut** executable:
`C:\Program Files\ProfileUnity\FlexApp\CTL\JitCut.exe`

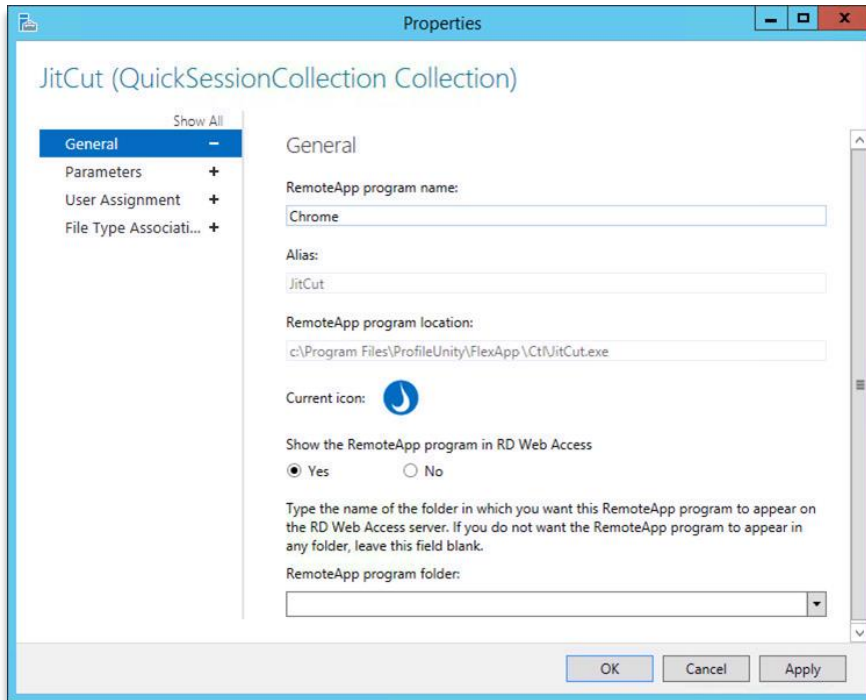




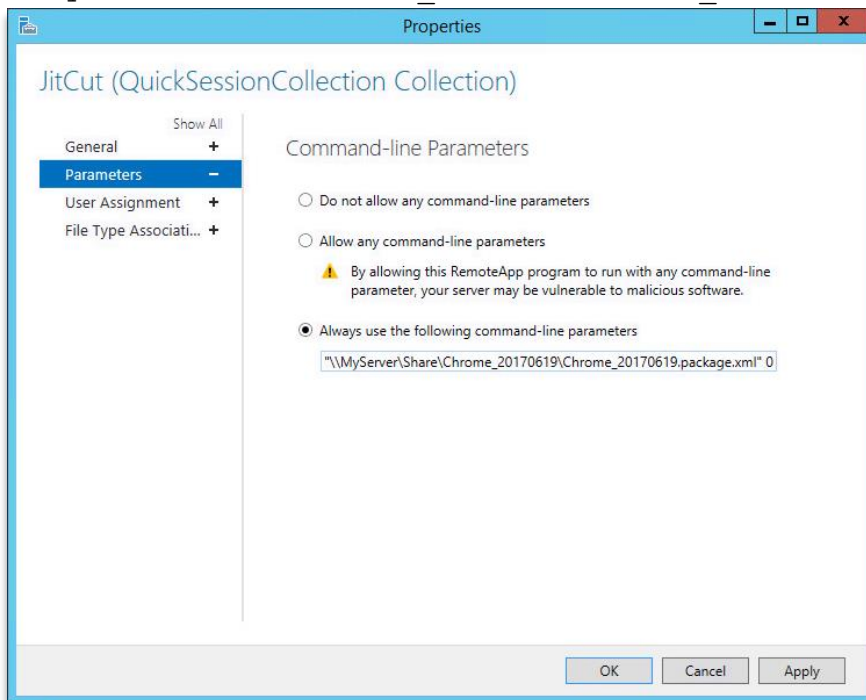
5. Complete the Publish RemoteApp Programs wizard.
6. Right-click on the new JitCut application and select **Edit Properties**.



7. On the **General** page, change the **RemoteApp program name** field to the name of the application intended to be published.

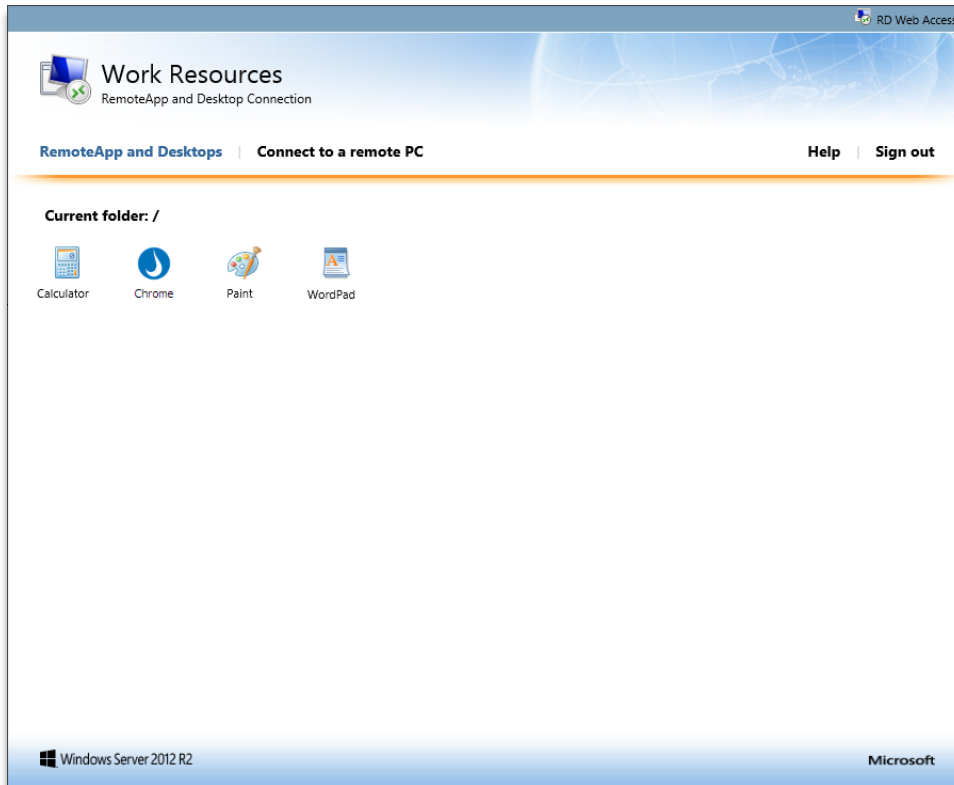


8. On the **Parameters** page, specify the path to the package xml (stored alongside the package FlexApp Packaging Console creates) using the following format:
"`\\MyServer\Share\Chrome_20170619\Chrome_20170619.package.xml`" 0



9. Press **OK** to complete the application configuration process.

10. Configuration is complete. The application can now be assigned and launched from the RemoteApp portal, the same as any other application.



Note: PowerShell can be used to customize the icon seen in the RemoteApp portal. To do so, launch a PowerShell prompt on the RemoteApp server and:

1. Type the following to replace the Application icon with a new one from a .ico, .exe or .dll:
`Get-RDRemoteApp -Alias "JitCut" | Set-RDRemoteApp -IconPath "C:\MyIcons\chrome.ico"`
2. Logout of RemoteApp portal and then back in for changes to take effect.