



**ProfileUnity™
with FlexApp™ Technology**

***FlexApp Integration with
Amazon AppStream***

Introduction

This guide has been authored by experts at Liquidware in order to provide information and guidance concerning the installation and configuration of ProfileUnity with FlexApp.

Information in this document is subject to change without notice. No part of this publication may be reproduced in whole or in part, stored in a retrieval system, or transmitted in any form or any means electronic or mechanical, including photocopying and recording for any external use by any person or entity without the express prior written consent of Liquidware Labs.

Liquidware Labs, Inc.

3600 Mansell Road

Suite 200

Alpharetta, Georgia 30022

U.S.A.

Phone: 678-397-0450

www.liquidware.com

©2020 Liquidware Labs Inc. All rights reserved. Stratusphere, ProfileUnity, FlexApp, FlexDisk and ProfileDisk are trademarks of Liquidware Labs. All other products are trademarks of their respective owners. 20-0707.6.8.3 R2

Contents

- AMAZON APPSTREAM AND FLEXAPP OVERVIEW 3**
- FLEXAPP AND APPSTREAM WORKFLOW 5**
- PREPARATION..... 5**
- IMPLEMENTATION PROCESS CHECKLIST 6**
- SETUP AMAZON S3 STORAGE 7**
- DOWNLOAD PROFILEUNITY WITH FLEXAPP 7**
- SETUP THE PROFILEUNITY MANAGEMENT CONSOLE 7**
- INSTALL THE PROFILEUNITY CLIENT INTO AMAZON APPSTREAM 8**
- REGISTER THE PROFILEUNITY AGENT WITH AMAZON APPSTREAM 9**
- FLEXAPP PACKAGING GUIDANCE (AS2 SHARED IMAGE OPTION) 10**
- FLEXAPP PACKAGING GUIDANCE (TRADITIONAL) 12**
 - FLEXAPP PACKAGING..... 12
 - ABOUT THE PACKAGING VIRTUAL MACHINE..... 12
 - TARGET VMs/POOL/FLEET 13
- BEHIND THE SCENES GUIDANCE WITH FLEXAPP..... 13**
 - HOW FLEXAPP APPLICATIONS SHORTCUTS ARE SENT TO APPSTREAM 13
- USING PRE-PACKAGED LIQUIDWARE APPS 14**
- CREATING YOUR OWN FLEXAPP PACKAGES..... 14**
- ASSIGNING FLEXAPP PACKAGES TO USERS 15**
- STORAGE CONSIDERATIONS 16**
 - FLEXAPP AND TRADITIONAL WINDOWS STORAGE 16
 - FLEXAPP AND S3 16
 - FLEXAPP AND FSX FROM AMAZON..... 16
- GETTING HELP INSTALLING PROFILEUNITY..... 18**
 - USING ONLINE RESOURCES..... 18
 - CONTACTING SUPPORT 18

Amazon AppStream and FlexApp Overview

“Amazon AppStream 2.0 is a fully managed application streaming service that provides users with instant access to their desktop applications from anywhere. AppStream 2.0 manages the AWS resources required to host and run your applications, scales automatically, and provides access to your users on demand. AppStream 2.0 provides users access to the applications they need on the desktop device of their choice, with a responsive, fluid user experience that is indistinguishable from natively installed applications.” - AppStream Description from the Amazon AppStream 2.0 Developer Guide “What is Amazon AppStream 2.0?”

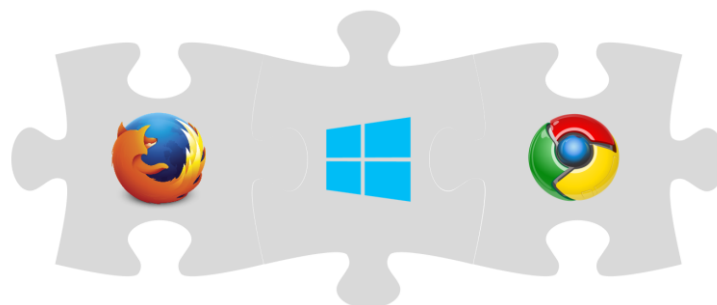
Amazon AppStream is a ground-breaking, fully managed application streaming service that delivers desktop applications from AWS to any end point running an HTML5 compatible browser. AppStream 2.0 provides robust, scalable publishing and hosting capabilities from the cloud. Application binaries are installed natively within the AppStream image builder instances for presentation to the corresponding users from an HTML 5 browser.

However, image management within AppStream can become cumbersome over time based on either the number of supported images or the number of natively installed applications. Integrating ProfileUnity FlexApp with AppStream alleviates these image management challenges by reducing the need for natively installed applications within the image builder instances. FlexApp redirects the natively application installs into virtual disks that essentially make the applications portable. Essentially each of the FlexApp packages are attached like puzzle pieces during the boot process of the image builder instance, user login, or a new post-login Click-to-Layer feature process.

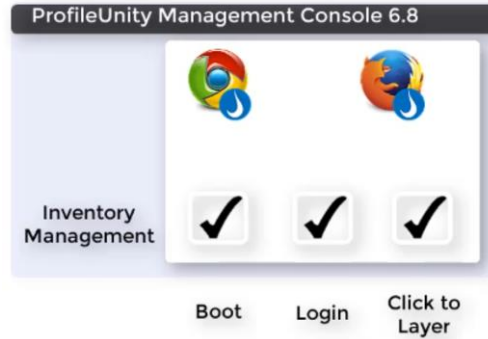
ProfileUnity’s FlexApp Technology is a fully managed solution geared towards optimizing the application lifecycle management process for enterprises. At its core, FlexApp simply redirects the native install of windows-based applications into virtual disks. At which point, the virtual disks are portable in nature and can follow users across devices and platforms. These virtual disks have been traditionally stored on windows SMB or CIFS based volumes. ProfileUnity and FlexApp now support cloud-based storage from the major platform vendors, including but not limited to Amazon S3, Microsoft Azure Blob, and Google Cloud Storage (GCS).

If you think of the Windows operating system like a puzzle piece, FlexApp packages attach like puzzle pieces during the user login or system boot process on the endpoint. A filter driver, much like a traffic cop directs communications between the corresponding FlexApp packages and the host operating system. Unlike, other solutions like application virtualization, the host operating systems thinks that the FlexApp packages are natively installed applications based on the seamless effective communication with the filter driver. However, the FlexApp packages are only attached through the traditional Microsoft virtual disk processes embedded within the core operating system.

At user logoff, the FlexApp packages are simply detached like puzzle pieces to round out the lifecycle experience for both administrators and users. This process drastically reduces the amount of clutter caused by traditional application installs and subsequent removal from the host operating system.

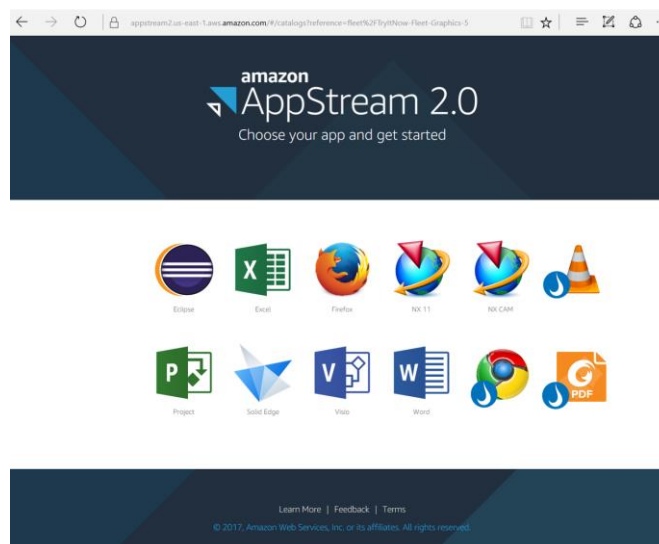


The delivery of FlexApp packages is managed from within the ProfileUnity Management Console. FlexApp packages are typically delivered through integration with Microsoft Active Directory user or group assignments. In an effort to further optimize the users login experience, FlexApp packages can be configured for layering into the environment at the following times:



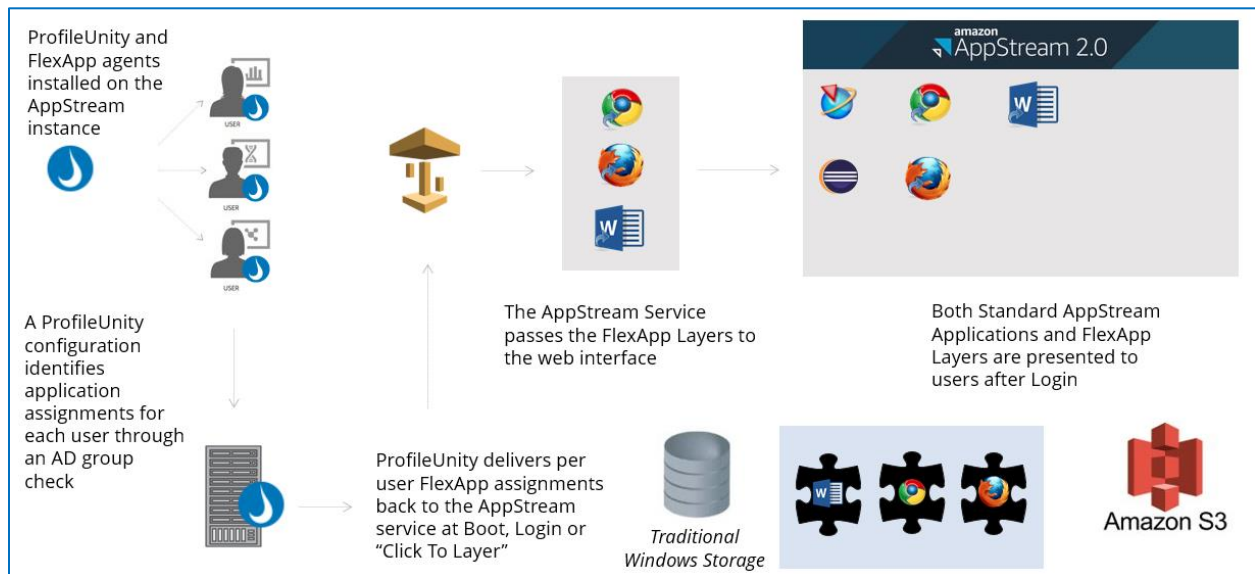
- **System Boot** – The individual FlexApp packages are attached during the system boot process. This is typically reserved for more complex applications. This also falls in line with standard application hosting best practices and guidance.
- **User Login** – The individual FlexApp packages are attached in the background as the user logs into the Windows endpoint
- **FlexApp Click-to-Layer** – For applications that are not typically needed right away during the user’s login, the FlexApp Click-to-Layer method can be leveraged. Essentially, an icon file with XML information is delivered to the user’s desktop during the login process. When the user double clicks this icon to activate the application, the XML makes a call to the corresponding FlexApp package in the background, attaches it to the host operating system, swaps out the temporary icon on the desktop for the official one, and launches the layered application from within the FlexApp package.

Ultimately, as seen in the graphic below, the integration of FlexApp with AppStream is seamless to the users during the AppStream login process. Both natively installed applications and the corresponding layered applications from the FlexApp packages are presented to them in the same, traditional AppStream interface. FlexApp package layering has no effect on the overall user experience AppStream users have come to expect.



FlexApp and AppStream Workflow

The following diagram represents the basic process of integrating FlexApp Layers within the AppStream platform.



Please explore the following resources for additional information:

Video: [FlexApp and Amazon AppStream Working Together](#)

Video: [Implementation Best Practices for FlexApp and AppStream](#)

Video: [ProfileUnity and FlexApp 6.8 Version Release Overview](#)

Preparation

Integrating ProfileUnity FlexApp with Amazon AppStream will require the following:

1. Acquire access and credentials to your Amazon S3 storage, Amazon WorkSpaces and Amazon AppStream account credentials. Amazon S3 storage will be utilized for the ProfileUnity configuration file(s) and FlexApp packages.
2. Acquire AD Domain Admin access and access to the NETLOGON folder of your domain controllers. Verify you have permission to create AD Group Policy Objects.
3. For Amazon AppStream,
 - a. The AppStream fleet must be domain joined.
 - b. The user must be authenticating to AppStream with a domain account via a company-managed SAML gateway.
4. Download the ProfileUnity with FlexApp software and documentation. You will need to use ProfileUnity with FlexApp Version 6.8.1 or higher.
 - a. Download the software from <https://www.liquidware.com/download>.
 - b. Download [Using Cloud Storage with ProfileUnity](#).
 - c. Download the [Installation and Configuration Guide](#) and review system requirements.
 - d. Download the [FlexApp Packaging Console Manual](#) and review system requirements.

- e. (Recommended) Download [FlexApp Packaging Console Guidance and Best Practices](#).
 - f. (Recommended) Download [Base Image Creation Best Practices](#).
5. For the FlexApp Packaging Console:
- a. The FlexApp Packaging Console should not be installed on the same machine as the ProfileUnity Management Console.
 - b. Each FlexApp Package or Layer should be created on a machine running the same operating system with the same bit level (32- or 64-bit) as the user machines on which it will be deployed.
 - c. Leverage standard on premises application packaging infrastructure that can connect to the ProfileUnity Management Console and to your S3 bucket (PRIMARY)
 - d. Currently, snapshots are not available within AWS. You can install the FlexApp Packaging Console on a standalone AppStream Image builder instance.

Implementation Process Checklist

The ProfileUnity FlexApp Packaging Console requires the following for installation:

Task	Status
Setup Amazon S3 Storage	
Downloaded ProfileUnity with FlexApp	
Setup the ProfileUnity Management Console	
Install the ProfileUnity Client into Amazon AppStream	
Register the ProfileUnity Client with Amazon AppStream	
Establish FlexApp packaging environment <ul style="list-style-type: none"> • Configure FlexApp Packaging Console AS2 Shared Image from Liquidware (Optional) • Setup the FlexApp Packing Console (Traditional) 	
Use Available FlexApp Packages or Create Your Own	
Assign FlexApp Packages to Users with the ProfileUnity Management Console	

Setup Amazon S3 Storage

Use the instructions given in [Using Cloud Storage with ProfileUnity](#) to configure your Amazon S3 cloud storage for ProfileUnity. These instructions walk you through creating a storage bucket with folders and access accounts for ProfileUnity and FlexApp. After completing these instructions, you will be able to install and configure ProfileUnity to make use of your new cloud storage bucket.

Download ProfileUnity with FlexApp

Download the ProfileUnity with FlexApp software from <https://www.liquidware.com/download>. This must be version 6.8.1 or higher.

Setup the ProfileUnity Management Console

Using the instructions given in the [Installation and Configuration Guide](#), install the ProfileUnity Management Console. Please see the guide for system requirements.

ProfileUnity's Guided Configuration Wizard offers the option to automatically configure your Activity Directory environment for you. You must be logged in as an AD admin for this to work. When you run the Guide Configuration Wizard, pick the "AWS AppStream utilizing S3 Storage" template so you may enter your ProfileUnity Console and ProfileUnity Client cloud credentials and the cloud storage paths created on your Amazon S3 cloud storage.

If you choose to manually configure your Active Directory settings, you will need to enter your ProfileUnity Console and ProfileUnity Client cloud credentials manually.

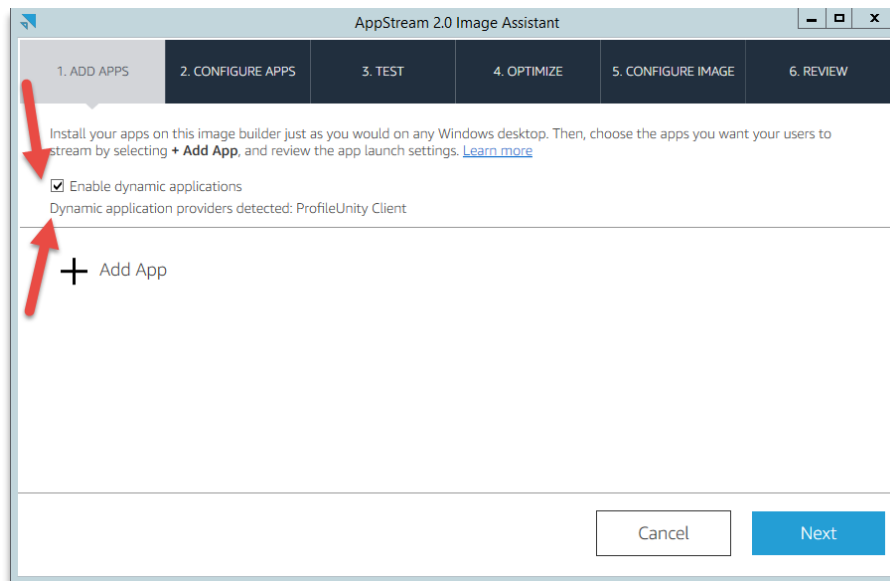
1. From the ProfileUnity Management Console navigate to **ADMINISTRATION > SETTINGS > CLOUD STORAGE**.
2. Click **Add Credentials** and enter the credentials for the ProfileUnity Console cloud access.
3. Click **Save** when done.
4. Click **Add Credentials** and enter the credentials for the ProfileUnity Client cloud access.
5. Click **Save** when done.
6. Click the blue **Update** button.
7. Follow the instructions in the "Manually Preparing Your Environment to Use the ProfileUnity Client" section of the [Installation and Configuration Guide](#) paying close attention to computer and user settings in the GPO. This is where you will complete mapping your cloud storage paths and cloud account credentials.
8. Make sure the following GPO settings are configured:
 - a. Enabled
 - b. INI File Path
 - c. Amazon S3 Storage Credential

Install the ProfileUnity Client into Amazon AppStream

Start the image builder instance with the MOST current image and AppStream agent to be used with ProfileUnity FlexApp and connect as an administrator.

1. Login to your ProfileUnity Management Console from your image builder instance. <https://your-server:8000>
2. In the upper right-hand corner of the ProfileUnity Management Console, go to the user name and select **Administration** from the drop-down list.
3. On the default **Settings** Tab, scroll down to the **ProfileUnity Tools** section.
4. Click on **Deploy Client Tools**.
5. Click **Yes** to confirm.
6. Select "Download" as your **Platform**, and click on the **Deploy** button. This will download the Zip file through your browser.
7. Extract **client-tools.zip** to client-tools directory.
8. Install the ProfileUnity Client agent software on your AppStream Image.
 - a. Right click **client-tools\LwL.ProfileUnity.Client.Startup.exe** and select "Run as administrator".
9. Press any key to continue.
10. Install All runtimes to the image
 - a. Liquidware's bundled Runtime installer. This provides a common set of libraries for FlexApp operations. Please read the following KB on using the [Runtime Installer Package](#) and for a link to the latest version of the installer.

Register the ProfileUnity Agent with Amazon AppStream



1. Open the Image Assistant on the image builder.
 - a. Check the option for **Enable dynamic applications**.
 - b. No need to add any apps unless you have installed apps in the Image Builder Instance that you want to publish to all users outside of ProfileUnity/FlexApp.
2. Configure apps
 - a. Click **Next**.
 - b. The configure apps step is not required for FlexApp and AppStream to work together. Click **Next**.
3. Test
 - a. Click **Next**.
4. Optimize
 - a. Click **Launch**.
5. Configure your image and check the option to use the latest agent.
6. Click **Disconnect and Create Image**.
7. You can now build your user Fleet based on this image.

FlexApp Packaging Guidance (AS2 Shared Image Option)

Amazon AppStream has released a new feature called Shared Images. These shared images are intended to increase efficiencies within the AppStream management process by building out repeatable processes or templates.

Liquidware is excited to create a shared image as part of this workflow based on our FlexApp Packaging Console. Enterprises can now use the Liquidware shared Image for packaging applications. Liquidware has created a fully configured FlexApp Packaging Console image with all the best practices already applied.

1. To get access to this shared image you will need to send information along so that this image can be shared with your AWS account.
 - a. Email AS2FPCImage@liquidware.com with the following information
 - b. Name:
 - c. Company:
 - d. AWS Account number:
 - i. How to find your AWS account number
https://docs.aws.amazon.com/IAM/latest/UserGuide/console_account-alias.html
 - e. AWS AppStream Region:
2. Once you have been given the confirmation the image was shared with your AWS account in the requested region you can start the process of building a fleet from this image.
 - a. Go to your AppStream console in the correct region.
 - b. While in AppStream, go to **Images** and **Image Registry**.
 - c. Change filter dropdown from **All Images** to **Shared with Me**.
 - i. You should see this image:
Liquidware_FlexApp_Packaging_Console_Base_OSversion_v6.8.x_v1.x.x
 - ii. OSversion should match your fleet, all OS versions will show under all images shared with me.
 - iii. v6.8.x will show the version of the FlexApp packaging console shared with you; this should match the version of your ProfileUnity Install.
 - iv. v1.x.x is the version based on any new changes to the image itself.
 - d. Once you see this image you can move to image builder and launch image builder.
 - e. Change All images to **Shared with Me**.
 - f. Pick **Liquidware_FlexApp_Packaging_Console_Base_OSversion_v6.8.x_v1.x.x** from the list.
 - g. Build your image as normal.
 - h. For the image builder steps this part does not have to be part of the domain.
 - i. Once your image is running you would login one time into the FlexApp Packaging Console to make sure it's connecting to the ProfileUnity Console.
 1. One you login successfully, close the FlexApp Packaging Console and edit the following XML to make the packager experience easier. This way the URL to the ProfileUnity console will be pre-filled in and so will their user name.
 1. Open **C:\Program Files (x86)\Liquidware Labs\FlexApp Packaging Console\FlexApp Packaging Console.exe.config.xml**

Change:

```
<PackageManagementUsernameRaw>jason</PackageManagementUsernameRaw>
```

To:

```
<PackageManagementUsernameRaw>%username%</PackageManagementUsernameRaw>
```

- j. Build your Packaging fleet from this new image with Domain join.
 - i. Click on **Images**.
 - ii. Click on **Image Builder** and look for the image you just created based on our FlexApp Packaging Console image.
 - iii. Create a fleet from your new image. Note: The user must be an administrator over the fleet VM's to package applications.
 - a. Platform: The OS version you picked from
 - b. Instance Family: General Purpose, Compute Optimized, Memory Optimized
 - c. Apps Included: FlexApp Packaging Console, Explorer and Firefox
 - d. Instance type should be a minimum of 8 GB of RAM
 - e. Active Directory is required
 - f. Finish out the rest of the settings to create your fleet from your version of the Liquidware shared image.

FlexApp Packaging Guidance (Traditional)

The FlexApp Packaging Console builds the individual application packages that can be layered into a user's environment. The goal is to provide a clean packaging environment, so that when packages are layered in, they include everything the application needs to run including all file and/or application dependencies. We want to create a successful, repeatable installation process across all user's desktops. Therefore, the FlexApp Packaging Console has a few system requirements including:

- The packaging console should be installed on a Windows operating system that can be reset after each package is created. Using a non-persistent pool or a virtual machine with a snapshot that can be rolled back is fully acceptable.
- Do not install the packaging console on the same machine where the ProfileUnity Management Console was installed.
- The FlexApp Packaging Console has an additional bit-level requirement. The bit-level of the operating system on which you package applications with the FlexApp Packaging Console needs to match the operating system of the desktop to which you deploy applications. If you are packaging applications to be used in a 32-bit environment, you must use the Packaging Console on a 32-bit OS. If you are packaging applications to be used in a 64-bit environment, you must use the Packaging Console on a 64-bit OS.

FlexApp Packaging

Follow all traditional or enterprise application packaging best practices in combination with a few additional FlexApp based recommendations

- FlexApp Packaging Console Best Practice Documentation
 - [FlexApp Packaging Console Best Practices](#)
- FlexApp Base Image Creation Best Practices documentation
 - [FlexApp Base Image Creation Best Practices](#)
- Install All runtimes to the image
 - <https://www.dropbox.com/s/pceifz9ru9m7e5j/FrameWorks.zip?dl=1>
Extract FrameWorks.zip and run "All.bat" as administrator.

About the Packaging Virtual Machine

Following application packaging best practice is critical to your success. The main goal with the FlexApp Packaging VM is to keep it as clean as possible between capturing applications and the make sure the OS is free from noise. You don't want anything left behind from a previous application installation and capture. Just uninstalling applications will not work; it must be as if they had never been installed.

- Disable Windows Updates
- Disable Volume shadow copy
- Disable any other service/software that might try and update themselves.
- Install All .Net runtimes
- Install All C++ runtimes
- Install the FlexApp Packing console
- Login to the FlexApp Packaging console
- Power down
- Snap shot the VM, this will be the snap shot point we will roll back to after each capture.
- Power on
- Login to the FlexApp Packaging Console

- Start and capture the app.

**Not following the C++ and .Net steps can cause large CPU and RAM usage from our agent.

Target VMs/Pool/Fleet

- Disable Windows Updates
- Disable Volume shadow copy
- Disable any other service/software that might try and update themselves while packages are active.
- Install All .Net runtimes
- Install All C++ runtimes

**Not following the C++ and .Net steps can cause large CPU and RAM usage from our agent.

Behind the Scenes Guidance with FlexApp

How FlexApp Applications Shortcuts are Sent to AppStream

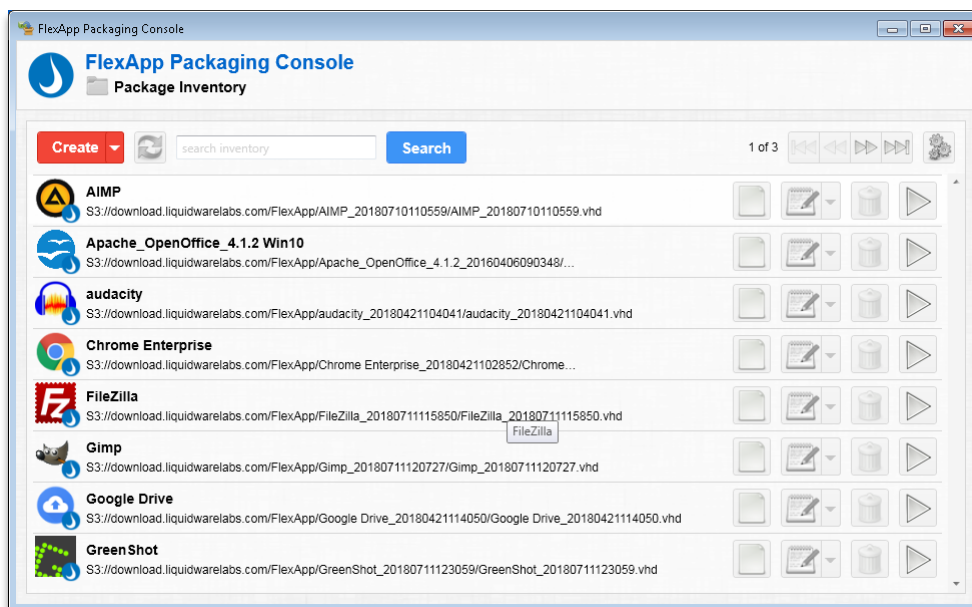
When you package an application, the application can place a shortcut in many locations and these shortcuts could be to a target file or binary that is or is not desired. Out of the box, here is the logic FlexApp will use to send shortcuts to the AppStream Web Interface.

- Any shortcuts in All Users Public Desktop override any shortcuts in All Users Start Menu programs. However, FlexApp will only send shortcuts that are targeting the following extensions *.exe, *.bat, *.cmd, *.vbs, *.ps1, *.msc.
- If there are NO shortcuts in the All Users Public Desktop path, then FlexApp will look to the All Users Start Menu programs.
- When FlexApp looks at the All Users Start Menu programs, it will only go one directory down. For example, if you have an app like Notepad++ and its path is Start Menu\Programs\Notepad++, only the shortcuts in Start Menu\Programs\Notepad++ will be sent to AppStream. Any shortcuts in Start Menu\Programs\Notepad++\help WILL NOT be sent to AppStream.
- On-boot applications will be active in the OS before a user session is created in most cases. When a new user session is seen, we send the on-boot applications to the AppStream web interface.

In the event during the packaging process you didn't edit, remove or change the shortcuts to your liking, that's ok. You can just use the FlexApp Packaging Console editor to edit the shortcut details post capture. Reference the [FlexApp Packaging Console Manual](#).

Using Pre-packaged Liquidware Apps

After logging in to the FlexApp Packaging Console, you will see the Package Inventory which lists all available FlexApp Packages. Starting with version 6.8.0, you will notice several FlexApp packages are automatically included in your inventory list. Liquidware has pre-packaged some popular applications and made them available to all users for immediate use via cloud storage. If a FlexApp package is a Liquidware App, the Liquidware Drop icon will be added to the application's icon on the lower right. Since the original packages were created and are owned by Liquidware, these FlexApps are subject to be updated, changed, or removed at any time. Therefore, it is recommended that the Liquidware Apps be cloned before extensive use in your own environment. As a general rule, the Liquidware Apps are intended for deployment to desktops running a 64-bit operating system.



For initial testing purposes, using a few of the Liquidware Apps may save time while setting up a Proof of Concept integrating ProfileUnity FlexApp with Amazon AppStream.

Creating Your Own FlexApp Packages

For more specific instructions on creating your own FlexApp Packages, please refer back to the [FlexApp Packaging Console Manual](#). The overall process following best practices will look like the following:

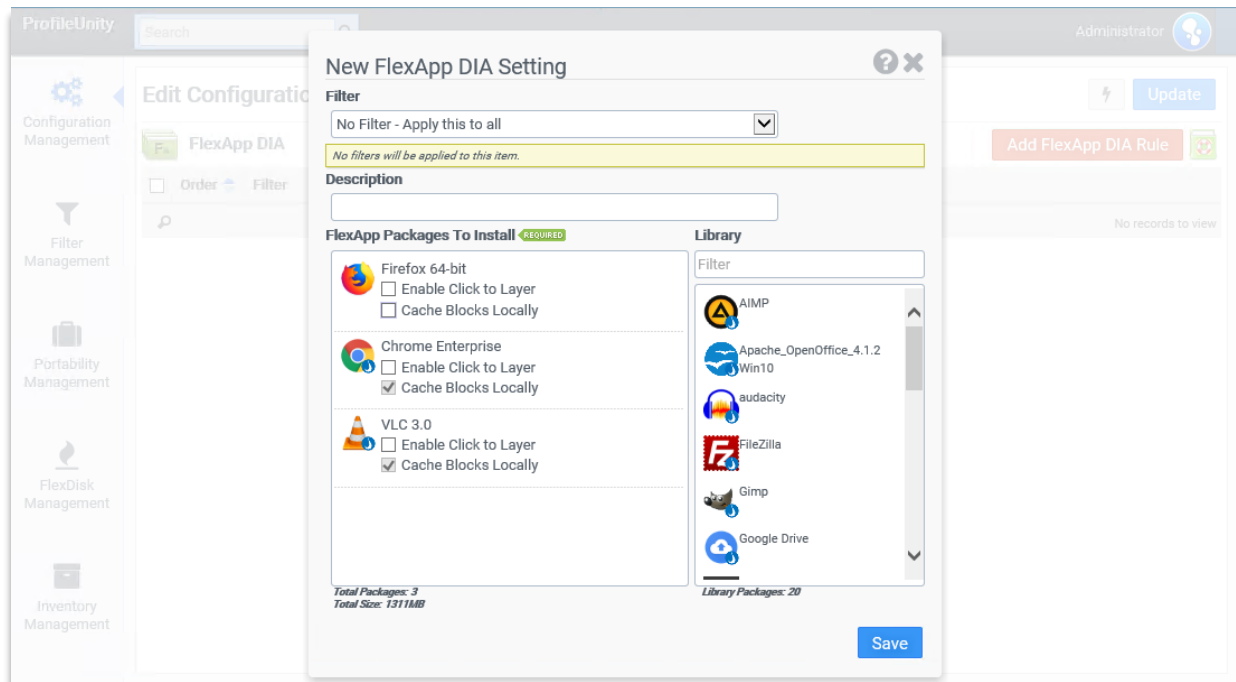
To package an application:

1. Power on the packaging console VM.
2. Login to the FlexApp Packaging Console.
3. Start and capture the application, creating an application package and saving it to your FlexApp cloud storage location.
4. Exit the FlexApp Packaging Console and power down the virtual machine.
5. Resort the virtual machine back to its initial snap shot of the FlexApp Packaging Console before packaging any applications.
6. Repeat these steps starting at step #1 until you are done packaging applications.

Assigning FlexApp Packages to Users

After applications have been packaged on the appropriate 32-bit or 64-bit OS platform using the FlexApp Packaging Console, admins switch back to the ProfileUnity Management Console for application delivery. The last step in delivering additional applications that are outside of the base image to users inside your environment is to use the power of ProfileUnity's configurable user environment management to selectively deploy your FlexApp application packages to just the users who need them. Use the instructions given in the "Enabling FlexApp DIA for Users in the Management Console" section of the *FlexApp Packaging Console Manual* to assign your FlexApp packages to your users.

It is during this assignment process inside the ProfileUnity Management Console that you can enable the Click-to-Layer feature for each FlexApp DIA package.



Storage Considerations

By design FlexApp can integrate with and bend around any storage design selection based on enterprise requirements. This is no different for the integration with the AppStream platform. Enterprise administrators can decide which storage approach makes sense for them and FlexApp can bend around those decisions. Here is some basic guidance around the various storage choices

FlexApp and Traditional Windows Storage

The fastest storage option for FlexApp packages is often traditional Windows file shares, because the enterprise administrator can just leverage a share directly from a server that is readily available in the same EC2 region. For some enterprises, this approach might be enough for a production rollout depending upon requirements. As FlexApp integrates with Amazon AppStream, those requirements often expand beyond the capabilities of a traditional Windows file share. At which point, the enterprise can consider some of the options below.

FlexApp and S3

Cloud Object Storage with Amazon S3 is clearly the standard best practice moving forward based on the durability, security, redundancy and management included by default.

“Amazon Simple Storage Service (Amazon S3) is the largest and most performant, secure, and feature-rich object storage service. With Amazon S3, organizations of all sizes and industries can store any amount of data for any use case, including applications, IoT, data lakes, analytics, backup and restore, archive, and disaster recovery. Amazon S3 is designed for 99.999999999% durability to protect data from site-level failures, errors, and threats, so that it is available to your end users and applications at all times.” - [Amazon S3 Overview](#)

Here are more references for additional information:

- [Amazon S3 Getting Started Guide](#)
- [Creating S3 buckets](#)
- [S3 Permission Guidance](#)
- [Using Cloud Storage with ProfileUnity](#)

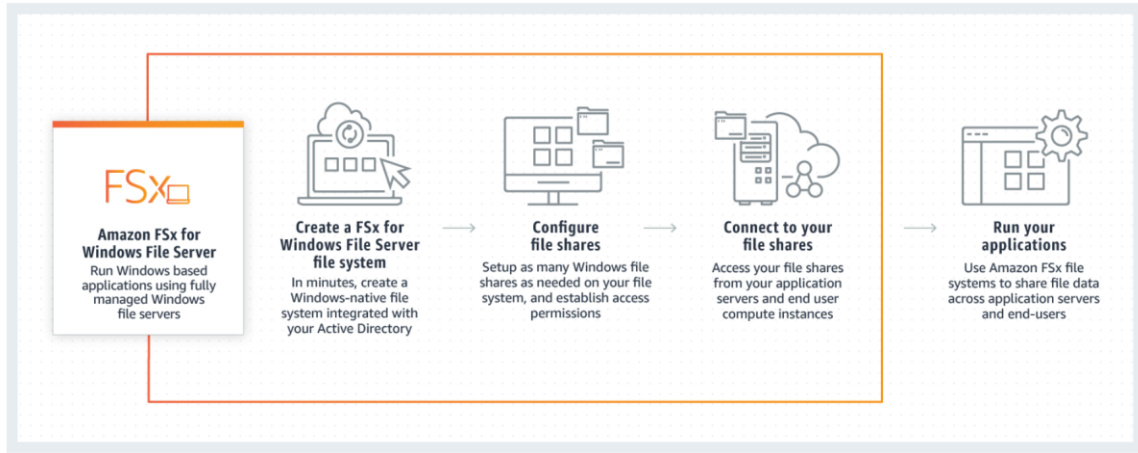
FlexApp and FSx from Amazon

Another intriguing option for cloud-based storage within Amazon is the new FSx platform. This allows the enterprise to essentially host traditional SMB type file shares in the cloud.

“Amazon FSx provides fully managed third-party file systems. Amazon FSx provides you with the native compatibility of third-party file systems with feature sets for workloads such as Windows-based storage, high-performance computing (HPC), machine learning, and electronic design automation (EDA). You don’t have to worry about managing file servers and storage, as Amazon FSx automates the time-consuming administration tasks such as hardware provisioning, software configuration, patching, and backups. Amazon

FSx integrates the file systems with cloud-native AWS services, making them even more useful for a broader set of workloads.” - Amazon FSx Overview

How it works



FlexApp packages simply integrate with the FSx file shares the same as with traditional on-prem Windows file shares. Only now, you have the robust, state-of-the-art Amazon scalable infrastructure behind them.

Getting Help Installing ProfileUnity

If you have questions or run into issues while installing and configuring ProfileUnity with FlexApp, Liquidware is here to help. Our goal is to provide you with the knowledge, tools, and support you need to be productive.

Using Online Resources

Liquidware maintains various kinds of helpful resources on our [Customer Support Portal](#). If you have questions about your product, please use these online resources to your full advantage. The Support Portal includes product forums, a searchable Knowledge Base, documentation, and best practices among other items. You can visit our website at <https://www.liquidware.com/support>.

Contacting Support

If you wish to contact our Support staff for technical assistance, please either log a request on the [Liquidware Customer Support Portal](#) or give us a call. Prior to Logging a Case you may want to review these helpful tips:

- Check the Product Documentation included with your Liquidware Product.
- Try to see if the problem is reproducible.
- Check to see if the problem is isolated to one machine or more.
- Note any recent changes to your system and environment.
- Note the version of your Liquidware product and environment details such as operating system, virtualization platform version, etc.

To speak directly with Support, please use the following numbers:

Main Line:	1-678-397-0460
Toll Free in US & Canada:	1-866-914-9665
Europe/Middle East/Africa:	+44 800 014 8097
Toll Free in Europe	
UK:	0800 014 8097
Netherlands:	0800 022 5973
Switzerland:	0800 561 271