



ProfileUnity™ with FlexApp™ Technology

Best Practices for the FlexApp Packaging Console

Release 6.8.7
November 20, 2024

This guide has been authored by experts at Liquidware in order to provide information and guidance concerning 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 2000
Alpharetta, Georgia 30022
U.S.A.

Phone: 678-397-0450

Web: www.liquidware.com

© 2024 Liquidware Labs Inc. All rights reserved. Stratusphere, CommandCTRL, ProfileUnity, FlexApp, FlexDisk, ProfileDisk, and FlexApp One are trademarks of Liquidware Labs. All other products are trademarks of their respective owners.

Table of Contents

General Best Practices	1
FlexApp Packaging Prerequisites	2
Building FlexApp Packaging Console Virtual Machines Checklist	3
FlexApp Packaging Optimization Checklist	4
Optimizing the FlexApp Packaging Console	5
FlexApp Layer Storage Options	8
FlexApp Layer Storage Replication Options	10
Microsoft DFS Considerations	10
Third-Party Storage Replication Technology	10
FlexApp Layer Testing	11
Additional Testing Steps	14
Repeatability Options	15
FlexApp Layer Troubleshooting	16
Handling Dependent Applications	18
Dependent Apps Excluded from FlexApp Layer	18
Dependent Apps Included within FlexApp Layer	18
Updating or Patching Application Layers	20
Key Questions	20
Getting Help	21
Using Online Resources	21
Troubleshooting with the Software	21
Contacting Support	21

General Best Practices

FlexApp Layer packaging best practices are built upon the same guidance leveraged by many typical standard application packaging platforms. Here are some common concepts to establish a baseline:

- Conduct initial target application analysis. Consider leveraging Application Assessment data from Stratusphere FIT, prior to the FlexApp packaging process.
- Maintain a centralized, reusable virtual machine (VM).
 - Virtualization technologies have evolved into a critical core component for any enterprise application packaging process.
 - Establish a range of FlexApp Packaging virtual machines representing the various operating system versions within the enterprise.
- Maintain a FlexApp Layer repository.
 - Leverage a centralized, highly available network share for FlexApp Package/Layer storage.
 - Additionally, where possible, leverage storage technology or best practices to ensure redundancy.
 - Ultimately this becomes critical with respect to any FlexApp Layer updates that need to occur within the production environment.
- Perform routine backups with respect to the FlexApp Layer repository.
- Perform preliminary application test installs as needed and document the results.

FlexApp Packaging Prerequisites

This section will discuss the benefits of preparation and planning with respect to the FlexApp Layer creation process. Often, one of the most overlooked aspects of this process is the staging required for the basic application installation, as part of the packaging process. What if the application has a heavy memory requirement? What if the disk footprint of the install is larger than most apps? These are questions that need to be answered before you start the packaging process, or else you run the risk of multiple failures. Here is a basic list of some of the items to be aware of:

- Packaging Operating System (OS) should match the OS of the target machines to which application layers will be deployed (i.e., Windows 10/11 or 2016/2019/2022).
- Review the [FlexApp Packaging Console Guide](#) for additional system requirements for your FlexApp Packaging VM.
- VM preferences
 - Establish hardware settings within the corresponding VMs that match or resemble production target application requirements.
- Identify prerequisite software that must be added to the FlexApp Package workspace.
- Identify application dependencies.
- Identify basic packaging steps based on outcome launch characteristics.
- Eliminate pop-ups.
- Eliminate license demand.
 - Identify application licensing requirements.
- Plan for any user customizations that will be required during the packaging process.
- Disable Anti-Virus on the FlexApp Packaging Console VM.
- Block GPO inheritance if the FlexApp Packaging Console is member of domain.
- Compare this list to additional knowledge base articles for FlexApp Best Practices on the [Liquidware Support Portal](#).

Building FlexApp Packaging Console Virtual Machines Checklist

There are many considerations that come into play when establishing a FlexApp Packaging Console VM. Below is a list of items for consideration.

Step	Task	Status
1	Determine which operating system versions are required (64-bit only). The OS of the packaging console should match the OS of the target machines to which application layers will be deployed. FlexApp requires Windows 10 (or Windows Server 2016) or higher.	
2	Maintain a minimum of one (1) VM with snapshots per OS version. Typically, applications should be packaged using the same OS version as the desktop to which packaged layers will be deployed, meaning you might need multiple packaging VMs.	
3	FlexApp Packaging Console desktops can be either physical or virtual. Typically, a VM would provide a more robust efficient use case.	
4	Review the software requirements contained in the <i>FlexApp Packaging Console Guide</i> .	
5	Configure Virtual Machine Preferences. Establish hardware settings within the corresponding VMs that match or resemble production target application requirements.	
6	Disable Anti-Virus on the FlexApp Packaging Console VM.	
7	Block GPO inheritance if the FlexApp Packaging Console is member of domain.	
8	Install the FlexApp Packaging Console on your VM. Follow the directions in the <i>FlexApp Packaging Console Guide</i> for downloading and installing the software . The installer will check to see if any prerequisite software is needed. Do NOT install the FlexApp Packaging Console on the same machine as the ProfileUnity Management Console.	
9	Compare this list to additional knowledge base articles for FlexApp Best Practices on the Liquidware Support Portal.	

FlexApp Packaging Optimization Checklist

There are also opportunities to optimize the FlexApp Packaging process and enhance the results. Preparation and planning with respect to the FlexApp Layer creation process are key. Often, one of the most overlooked aspects of this process is the staging required for the basic application installation, as part of the packaging process. What if the application has a heavy memory requirement? What if the disk footprint of the install is larger than most apps? These are questions that need to be answered before you start the packaging process, or else you run the risk of multiple failures. Below are some items to consider with respect to FlexApp Packaging optimization.

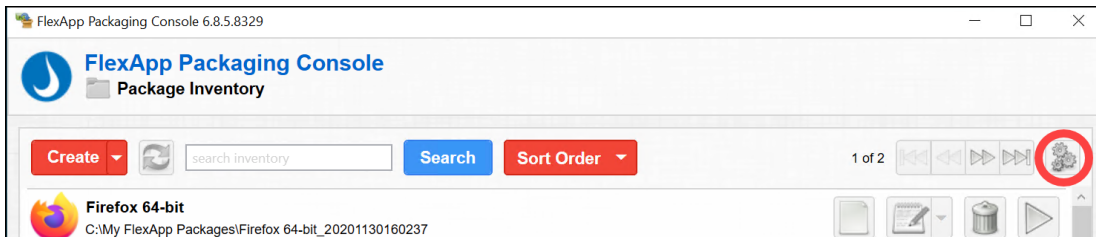
Step	Task	Status
1	<p>Select a preferred browser.</p> <p>For now, Google Chrome, Firefox, or recent versions of Internet Explorer are recommended to access either the FlexApp Packaging Console or the ProfileUnity Management Console</p>	
2	<p>Eliminate browser pop-ups.</p>	
3	<p>Identify application dependencies.</p> <p>If an application being packaged requires other software prerequisites, one package can be created to hold both the application and its prerequisites. Start by packaging the prerequisite software first. Then extend the package to include the original application to be packaged.</p>	
4	<p>Identify application licensing requirements.</p> <p>Eliminate license demand. Application licensing is done on the FlexApp Packaging Console. If your application binds itself to any information tying it back to the FlexApp Packaging Console, your application's license will fail when the application is opened on your target OSs. To work around this scenario, you would move to a centralized licensing model where the application license can be staged into the base image of your OS and the FlexApp Packaging Console. The other option if your application can silently auto-register licensing on startup would be to not disrupt the user each time with licensing.</p>	
5	<p>Identify basic packaging steps based on outcome launch characteristics.</p> <p>Plan for any user customizations that will be required during the packaging process.</p>	
6	<p>(Optional) Perform preliminary application test installs and document the results.</p>	
7	<p>Compare this list to additional knowledge base articles for FlexApp Best Practices on the Liquidware Support Portal.</p>	

Optimizing the FlexApp Packaging Console

To optimize the FlexApp Packaging Console, complete the following steps:

1. Open Explorer and navigate to C:\Windows\System32. Right-click **CompatTelRunner.exe** and click **Properties**.
 - a. Under the Security tab, click **Advanced**, followed by **Change Owner**.
 - b. Type in **Administrators** and click **OK** on all three windows.
 - c. Right-click **CompatTelRunner.exe** and click **Properties** again, Security tab. Highlight **Administrators** and click **Edit**.
 - d. Put a check by **Full Control** and click **OK**. Say **Yes** to the Security Warning dialog and click **OK** to close the Security and Properties windows.
 - e. Shift+Delete **CompatTelRunner.exe**, click **Continue** and **Yes** if prompted.
2. In similar manner, take ownership of "C:\Windows\SysWOW64\OneDriveSetup.exe"
 - a. Right-click, **Properties**, **Security**, **Edit** again.
 - b. Remove all existing ACL's and add in only one for Everyone with DENY for Full Control.
 - c. **OK** out of all Windows, clicking **Yes** to any prompts.
3. Open **Windows Security, Virus & threat**. Click **Manage** settings.
 - a. Turn Off all of the switches, clicking **Yes** if prompted.
4. Launch gpedit.msc.
 - a. Computer Configuration, Policies, Administrative Templates, Windows Components, Windows Update and Disable **Configure Automatic Updates**.
 - b. If the machine will be joined to a domain, also open Computer Configuration, Windows Settings, Security Settings, Local Policies, Security Options and Enable **Domain member: Disable machine account password changes**.
5. Launch Regedit.exe, delete the following Key:
HKLM\SYSTEM\CurrentControlSet\Services\DiagTrack
6. Open the shortcut to the FlexApp Packaging Console and connect to the corresponding address of your ProfileUnity Management Console. An address would typically be in the following format:
`https://servername:8000.`
7. Log in to the ProfileUnity Management Console with Administrator credentials.

8. Upon first logon to the FPC, you will be prompted to **Enable Rollback** which engages System Restore on Desktop OSes to maintain a clean state without needing access to the hypervisor for reverting to clean snapshots.
 - a. You can enable or disable Rollback at any time on the Settings screen.
9. You will then be prompted to run **Machine Optimizations**. This is to keep a quiet packaging environment and required prior to launching the FlexApp Packaging Console. Once complete, click **OK** and the machine will automatically reboot. Proceed to log back into the FPC.
10. Click the **Settings** button in the top right corner of the screen.



11. Enter information to streamline the FlexApp Layer packaging process.
12. (Optional) To save time, enter a **Default VHDX Location**. A highly available network storage location is recommended.

Settings

Use this screen to configure the FlexApp Packaging Console. When you're done, click OK.

General Information

Release 6.8.5.8329 e93546b8
Version 6.8.5.8329

Basic Settings

Default Installer Location
The default installer location to be used when creating a new package

Default VHDX Location
The default location to be used when creating a new VHDX package

S3 Cloud Credentials
The default credentials to be used when accessing the Cloud on S3

S3 Cloud Location
The default location to be used when creating a new Cloud package on S3

Azure Cloud Credentials
The default credentials to be used when accessing the Cloud on Azure

Azure Cloud Location
The default location to be used when creating a new Cloud package on Azure

Google Cloud Credentials
The default credentials to be used when accessing the Cloud on Google

Google Cloud Location
The default location to be used when creating a new Cloud package on Google

Default Local Install Location
The default location to be used when creating a new local package.

13. Configure any additional options.
14. Click **OK** when finished updating settings.
15. If you are going to use hypervisor snapshots, shut down the machine and take a clean snapshot now. Otherwise, the FPC is now ready.

FlexApp Layer Storage Options

FlexApp Layers can be stored on any available location that users have access to. One of the key features of FlexApp is its ability to seamlessly integrate with any infrastructure. Leveraging any existing storage platform is a good example of this. Below is some basic guidance to get started:

- How to determine storage requirements
 - After the FlexApp candidate list has been identified, calculate the basic install footprint required for each application.
 - These calculations can be tabulated to come up with the total disk footprint required for the corresponding FlexApp layers.
 - At which point, an N + 1 share can be created for redundancy.

Note: Stratusphere FIT can conduct an application assessment of existing applications and desktops to establish the basic install footprint applications.

- Simple
 - Leverage existing VDI best practices. Most VDI platforms call for a specific storage footprint during installation and configuration.
 - Creating shares off of this existing infrastructure is a very straight forward approach.
 - Most VDI platforms call for the creation of a Windows Server VM that acts as a storage appliance, hosting the various shares required.
 - Follow existing user permission guidance for the most simplistic approach.
- Advanced
 - Storage vendors like Nutanix, NetApp, and so on can be leveraged to host the FlexApp Layer repository.
 - Follow vendor specific installation and configuration guidance when creating the various file shares.
 - FlexApp will seamlessly integrate with and leverage these new file shares on the advanced hyper-converged platform.
- Cloud Storage
 - FlexApp packages can also be stored within AWS S3, Azure Blob and Google Storage platforms.

- Each platform has options for redundancy and replication for global accessibility.
- Disaster Recovery concepts
 - FlexApp Layer VHDX or VMDK files can be replicated across storage locations to achieve a basic DR strategy.
 - There are many options for replication including Microsoft DFS. Most storage vendors have replication software included with the solution and can be leveraged to achieve results.

FlexApp Layer Storage Replication Options

FlexApp Layer replication for enterprise scalability has become a very important question recently. Here are a few examples of how to achieve results with FlexApp leveraging known platforms available to the enterprise.

Microsoft DFS Considerations

- Microsoft DFS is a platform geared towards storage replication, and distribution of content within the enterprise environment.
- Domain Namespaces—Namespaces allow administrators to create single, logical reference points for folders that contain content.
- DFS Targets and Replication—These Namespaces can encompass multiple targets or locations. Creating multiple targets can provide greater redundancy for enterprise users with respect to application availability.
- FlexApp Layers can be organized within DFS infrastructure and replicated across sites using unique namespaces and multiple targets.

Third-Party Storage Replication Technology

Most storage vendors, whether it be hyper-converged offerings from Nutanix, or proprietary offerings from EMC or NetApp, include software designed to provide content replication benefits. FlexApp Layers can easily be stored on files shares hosted within these various storage platforms and then replicated across sites using the features provided by the storage platform.

FlexApp Layer Testing

The following are common FlexApp Testing techniques for your reference:

- **Primary Launch Test**

Perform a simple test launch of the completed FlexApp Layer prior to reverting the FlexApp Packaging Console VM. This will quickly identify the status of the target FlexApp Layer.

- **Secondary Launch Test**

- Leverage the same FlexApp Packaging Console VM.
- Revert to a prior snapshot essentially rolling back to a clean baseline.
- This is sometimes referred to a “Clean” test considering the corresponding application within the FlexApp Layer has never existed within the clean snapshot.
- This test provides important data with respect to real time deployment success capabilities of the FlexApp Layer.

- **Pre-Production Test**

- Leverage a test environment similar to production for policy and third-party software interoperability testing.
- This test provides critical data with respect to actual end-user experience using the FlexApp Layer.

- **Basic Application Launch Testing**

When the corresponding FlexApp Layer launches, perform simple usage tests to gauge application functionality:

- Click drop-down menus.
- Attempt to change application settings.

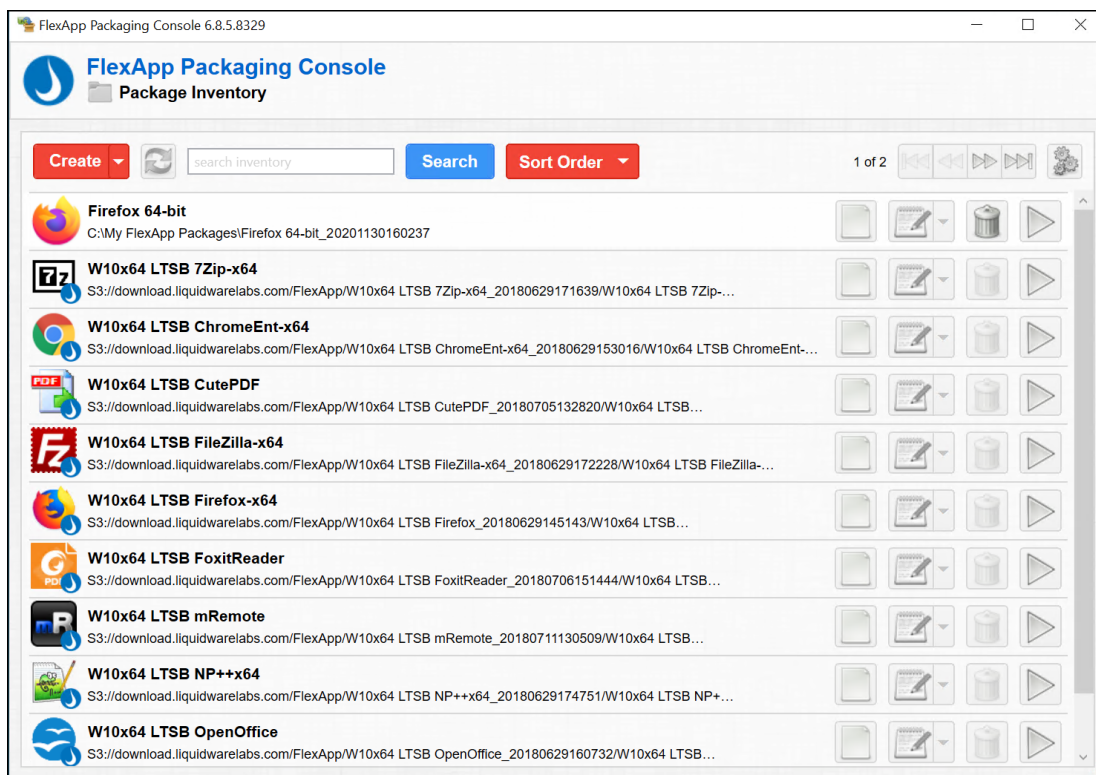
To start testing, complete the following steps:

1. From the FlexApp Packaging Console, open a web browser and connect to the corresponding address of your ProfileUnity Management Console, typically `HTTPS://servername:8000`. Log in to the ProfileUnity Management Console with your admin credentials.
2. Initial test
 - a. The final step of the FlexApp Layer packaging process uninstalls the application from the FlexApp Packaging Console.

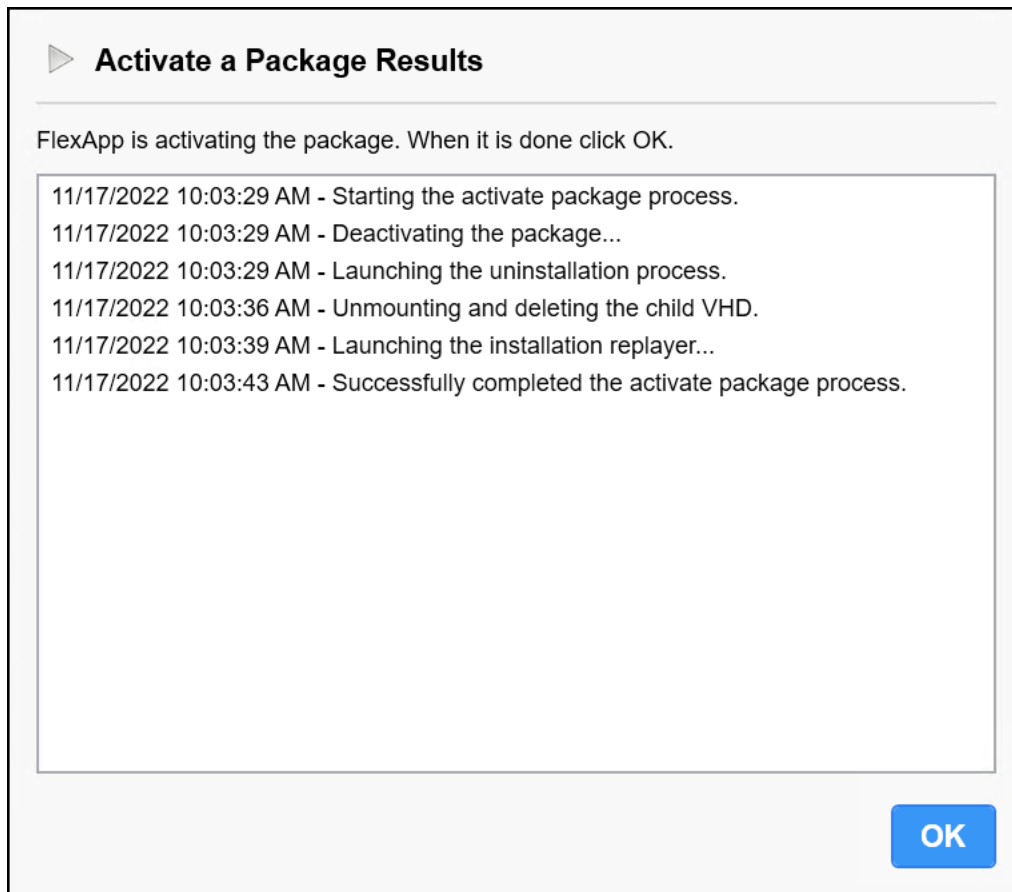
- b. Activate the new FlexApp Layer from the FlexApp Packaging Console and take note of the results.
- c. If the FlexApp Layer deployed successfully and PASSES initial launch testing, then you can move onto the CLEAN test phase.
- d. If the FlexApp Layer fails, then additional troubleshooting steps should be taken.

See the [FlexApp Layer Troubleshooting](#) section of this guide for additional details.


3. On the FlexApp Packaging Console screen, click the **Activate** button in the column on the right for the package you want to activate.



4. A popup window appears, showing the progress of the activation. Click **OK** in the popup window when the activation is complete.



As soon as the package is activated, the **Activate** button on the FlexApp Packaging Console screen is replaced by a **Deactivate** button.

5. To deactivate an application, click the **Deactivate** button. As was the case during the activation process, a popup window appears, this time showing the progress of the deactivation. Click **OK** when the deactivation is complete.
6. Secondary launch test
 - a. Revert the FlexApp Packaging Console VM to a clean snapshot. This eliminates any trace of the corresponding application.
 - b. Log in to the FlexApp Packaging Console.
 - c. Select the corresponding FlexApp Layer.
 - d. Click the  (**Activate**) button. This attaches the corresponding FlexApp Layer to the FlexApp Packaging Console.

Passing the CLEAN test means that the FlexApp Layer has proven functionality without any existence of the original install elements during the FlexApp Packaging process.

- If the FlexApp Layer PASSES the CLEAN launch test, then it is safe to proceed to UAT testing.
- If the FlexApp Layer FAILS the CLEAN launch test, then additional troubleshooting steps will be necessary.

See the [FlexApp Layer Troubleshooting](#) section of this guide for additional details.

7. UAT Acceptance Testing

- a. With the new FlexApp Layer added to the ProfileUnity Console's FlexApp Inventory, it can now be assigned to a 'test' configuration and deployed to a UAT or test environment that mimics production.
- b. Within the test configuration, FlexApp Module, assign the new FlexApp Layer. You may choose to use a Filter to ensure only certain users will receive the new layer for testing, or omit the filter so all UAT test users can see and test the new application.
- c. Deploy the configuration and logon to the UAT test environment and verify application functionality with similar policies and third-party software as the production environment.

Additional Testing Steps

App owner testing a simulated production pool

- Activate the application, open all shortcuts, and check licensing status.

User Acceptance testing (UAT)

- Assign the application, use it like normal, and check licensing status.

Repeatability Options

There are a few concepts within the virtualization space that allow for efficient repeatability of the application packaging process. All of these options allow for increased efficiency within the FlexApp packaging process, so there are no wrong choices when it comes to the repeatability approach you chose.

- Virtual Machine Snapshots
 - Embedded technology in most virtualization platforms that allows for the creation of a “photocopy” of the entire OS.
 - You can essentially revert or rollback the operating system to that “photocopy”.
 - You can easily maintain many different snapshots in various states to maximize efficiency.
 - Taking a snapshot after the FlexApp Packaging Console is configured would provide for an efficient repeatable process.
 - Additional snapshots could be leveraged for either large applications or with respect to troubleshooting opportunities.

- Non-Persistent Desktops
 - Embedded technology in most virtualization platforms that allows for a settings change within the corresponding virtual disk.
 - You can essentially set the virtual disk to a “non-persistent” state, after which any changes made within virtual machine would be discarded when you power off the VM.
 - After the FlexApp Packaging Console has been configured, setting the virtual disk to non-persistent would provide for an efficient repeatable process.

- VDI infrastructure can be leveraged with the FlexApp packaging process when available.
 - In addition to the options listed above, Administrators have the option to destroy the corresponding FlexApp Packaging Console at logoff after the FlexApp layer has been created.
 - The FlexApp packaging VMs can then be recreated and assigned to the FlexApp administrators.

FlexApp Layer Troubleshooting

When troubleshooting the FlexApp Layer, use the following workflow to help you identify what is causing the issue you are having.

- Does the app work correctly after capture? Attempt to launch the target application after install but before clicking the Finish button and closing the layer.
 - If no, stop here.
 - Check that the app works natively to be sure you do not have a corrupt installer or compatibility issue with the FlexApp Packaging machine, itself.
- Does the app work and does licensing check out after rolling back the FlexApp Packaging Console?
 - If no, stop here.
 - Document the error.
 - Check event viewer for errors or warnings that could apply to the application.
 - Some error messages have reference to:
 - If reg key
 - Try to export reg key from native install and import on the FlexApp Packaging Console as a test.
 - Is the reg key in the cap file but not being put down?
 - Is the reg key missing from the cap file?
 - Is the reg key being excluded by the virt config?
 - If file
 - Is the file in the virtual disk?
 - If file is missing from virtual disk, can you copy it in from native install and have the app work?
 - Service
 - Virt log should show installing services failure?
 - Printer
 - Virt log will show installing printer failure?

- Compare application behavior to native install.
- Does the application have drivers and were they installed properly upon activation?
- Does the app work on the endpoint or desktop pool outside of the FlexApp Packaging Console?
 - If there is a licensing error, then it is possible the corresponding licensing cannot be moved from machine to machine or might require additional steps to accomplish licensing compatibility.

Handling Dependent Applications

Sometimes target applications might need prerequisite software installed first. The FlexApp Packaging Console allows the packaging of applications separately or the packaging of applications into one FlexApp Layer. To package applications into one layer, the admin would package the dependent application first and then extend the FlexApp Layer to include another application, finally ending with the target application.

Dependent Apps Excluded from FlexApp Layer

1. Install the dependent application on the FlexApp Packaging Console OS before starting the packaging process. (For example, .NET or C# Runtimes)
2. Start the FlexApp Packaging Console and capture the target application. (For example, Paint.NET)
3. Save and finalize the target FlexApp Layer.
4. Revert to a clean snapshot on the FlexApp Packaging Console VM.
5. Start the FlexApp Packaging Console and capture the dependent application.
6. Save and finalize the target dependent FlexApp Layer.
7. Revert to a clean snapshot on the FlexApp Packaging Console VM.
8. Assign the new FlexApp Layer to a test environment for app owner testing.

The Paint.NET FlexApp Layer attempts to find the .NET dependency on the host OS during the launch process. If the dependency is not installed within the target environment, the application will not function properly.

Dependent Apps Included within FlexApp Layer

1. Start the FlexApp Packaging Console.
2. Start the packaging process.
3. Package the dependent application on the FlexApp Packaging Console. (For example, .NET)
4. Before finishing the capture, install the target application. (For example, Paint.NET)
5. Save and finalize the FlexApp Layer.

6. Revert to a clean snapshot on the FlexApp Packaging Console VM.
7. Assign the new FlexApp Layer to a test environment for app owner testing.

The Paint.NET FlexApp Layer leverages the .NET dependency inside the FlexApp Layer during the launch process.

For more information on extending, cloning, and patching FlexApp Layers, refer to the [FlexApp Packaging Console Guide](#).

Updating or Patching Application Layers

Just like any application packaging process, FlexApp Layers can be updated on an as-needed basis. The following guidance represents a basic quick start guide for updating FlexApp Layers. Most enterprises will simply integrate FlexApp Layering techniques into their existing workflows.

One common best practice technique is to make a clone of the corresponding FlexApp Layer before updating it. This is similar to taking a snapshot of a VM. Making a clone of the corresponding FlexApp Layer first provides an additional roll back capability within the update process.

Key Questions

- Is it easier to create a new FlexApp layer with the new version of the application?
- Is it easier to clone a FlexApp Layer and perform the update within this cloned Layer?

There are no wrong answers. Every application will present different requirements or best practices with respect to update strategy. Enterprise application packaging teams will often have best practices for both application packaging and updating that FlexApp Layering can integrate with.

For more information on extending, cloning, and patching FlexApp Layers, refer to the [FlexApp Packaging Console Guide](#).

Getting Help

If you have questions or run into issues while using our software, Liquidware is here to help. Our goal is to provide you with the knowledge, tools, and support you need.

Using Online Resources

Liquidware maintains various kinds of helpful resources on our [Customer Support Portal](#). If you have questions about your product, use these online resources. The Support Portal includes product forums and a searchable knowledge base, as well as the ability to submit a case to the Liquidware Support system on the [Liquidware Customer Support Portal](#). For product documentation, refer to our [Liquidware Document Repository](#).

Troubleshooting with the Software

ProfileUnity and FlexApp provide full logging capabilities to track activities. In addition, Liquidware provides several avenues for self-service, which includes the [Liquidware Customer Support Portal](#), [Knowledgebase](#), and [Online Training](#) site. You may find answers to many of your questions, along with Liquidware Self Service options.

Contacting Support

If you need to contact our Support staff for technical assistance, log a request on the [Liquidware Customer Support Portal](#). Prior to logging a case you should 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.