

ProfileUnity™
with FlexApp™ Technology

FlexApp One™ Applications

v6.8.5

Introduction

This guide has been authored by experts at Liquidware in order to provide information and guidance concerning the use of FlexApp One™ applications.

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

©2021 Liquidware Labs Inc. All rights reserved. Stratusphere, ProfileUnity, FlexApp, FlexDisk, ProfileDisk and FlexApp One are trademarks of Liquidware Labs. All other products are trademarks of their respective owners. 21-0824

Contents

WHAT'S NEW FOR FLEXAPP ONE? **4**

VERSION 6.8.5.63276 - RELEASED AUGUST 18TH, 2021 4

OVERVIEW **5**

FLEXAPP BENEFITS **5**

SOFTWARE REQUIREMENTS **6**

FLEXAPP ONE BUNDLER 6

FLEXAPP ONE APPLICATIONS 6

PREREQUISITES TO RUN FLEXAPP ONE APPLICATIONS 7

WELL-MANAGED DEVICES 7

LICENSING CONSIDERATIONS **7**

ABOUT FLEXAPP ONE APPLICATIONS **8**

NAMING 8

STORING 8

CLOUD SYNCING 8

SAMPLE USE CASE **8**

FLEXAPP ONE WORKFLOW 8

CREATING A FLEXAPP PACKAGE **9**

SETTING UP THE FLEXAPP ONE BUNDLER **9**

CREATING A FLEXAPP ONE APPLICATION FROM A FLEXAPP PACKAGE **9**

SAMPLE BUNDLER COMMAND LINE USAGE 10

ADDITIONAL BUNDLER OPTIONS 10

FLEXAPP ONE APPLICATION COMMAND LINE ARGUMENTS **12**

MICROSOFT ENDPOINT MANAGER DEPLOYMENTS **13**

PROCESS OUTLINE 13

DETECTION RULES 14

DEPLOYING THE FLEXAPP ONE SERVICE TO END USER MACHINES 14

DEPLOYING FLEXAPP ONE APPLICATIONS TO END USER MACHINES 15

MICROSOFT AVD, CITRIX, OTHER REMOTE DESKTOP DISTRIBUTION 16

ONEDRIVE DEPLOYMENTS 17

CONFIGURE ONEDRIVE GPO ADMINISTRATIVE SETTINGS: 17

CONFIGURE THE ONEDRIVE GPO 18

SAMPLE COMMAND LINE USAGE 18

SYSTEM CONTEXT OPTIONS 18

USER CONTEXT OPTIONS 19

USING LOGIN SCRIPTS 19

SUPPORT LOGS 19

GETTING HELP 20

USING ONLINE RESOURCES 20

CONTACTING SUPPORT 20

What's New for FlexApp One?

Version 6.8.5.63276 - Released August 18th, 2021

WHAT'S NEW

The first version of FlexApp One is released to market. FlexApp One makes offline distribution and consumption of applications possible.

KNOWN ISSUES/LIMITATIONS

- Installing or uninstalling applications into the operating system that could be intermingled with active FlexApp One applications could present challenges. Note that FlexApp One applications may have to be stopped in order to install or uninstall other software on the end user device. Support of installing or uninstalling applications into the operating system with active FlexApp One applications is best effort on well-managed devices.

- FlexApp One applications are incompatible with the simultaneous use of ProfileUnity on an end user device. To get around this limitation, please read the following KB article: How to Use FlexApp One and ProfileUnity Version 6.8.4.

Overview

ProfileUnity's FlexApp technology provides a way to consume applications without installing them within each workspace. FlexApp allows IT administrators to assign department-level applications to groups of end users. By making use of virtual disks for application storage, FlexApp enables organizations to create a single base image for all users and layer in extra applications as needed thus reducing storage requirements to maintain multiple base images for different departments or segments of users. FlexApp delivers applications which do not install into the workspace, thereby keeping the workspaces clean. Each FlexApp will be the same when deployed onto varied machines. Since no applications are installed with FlexApp, there is typically no need for complex uninstallation scripts to forcefully remove files, settings, or services.

Up until the release of FlexApp One, the ProfileUnity Management Console was required to run FlexApp packages and the user's workspace needed to be well connected to the corporate domain. With FlexApp One, IT admins now have an easy way to consume user applications on a variety of endpoint or desktop types. The FlexApp One bundler takes an existing FlexApp package and creates an executable that can be distributed in a variety of ways including via your Microsoft Corporate Portal, SCCM, Microsoft Endpoint Manager (formerly Intune), a flash drive, cloud storage like Dropbox, or OneDrive.

Using FlexApp One applications allows remote desktop or laptop users run applications when disconnected from the corporate network. IT admins can deliver FlexApp One applications to end users on clean, well-managed machines ahead of time. Then end users can run the applications locally when needed.

FlexApp One applications run on demand and act as if they are fully installed on the local machine. By default, when the workspace is restarted, the application is dropped and the machine remains as if the application had never been installed, leaving a clean base image or laptop.

IT admins can use both FlexApp One applications and traditional FlexApp packages for application delivery. FlexApp One applications fulfill the basic need to quickly distribute applications to end users. For advanced use cases, FlexApp packages can rely on the power of ProfileUnity for additional delivery options including cloud storage options, context-aware filters, privilege elevation, application restrictions, and registry fixups.

FlexApp Benefits

The following list highlights the core benefits of using FlexApp for application delivery:

- No file system tattooing (no Windows rot)
- No installation variances
- No complex uninstallations
- One-to-many centrally managed applications
- Simplifies application updates
- Drastically reduces image proliferation and management

Users of FlexApp One enjoy the following additional benefits:

- Supports offline desktop & laptops
- Supports remote workers
- Supports cloud desktops
- Portable executable makes distribution simple

- Seamless Microsoft Endpoint Manager (MEM) support
- Easy way to try FlexApp

Software Requirements

The software download for FlexApp One can be found with the ProfileUnity Documentation under **Tools & Resources**. This ZIP files includes:

- **FlexApp One Bundler** – bundles traditional FlexApp packages into distributable executables.
- **FlexApp One Documentation**
- **FlexApp Packaging Console Installer** – used to create a traditional FlexApp package. For more information, reference the *FlexApp Packaging Console Manual*. If the FlexApp Packaging Console will be used without being connected to the ProfileUnity Management Console, you need to create your FlexApp packages offline. See this KB article for more information on logging in to the FlexApp Packaging Console in offline mode.
- **FlexApp Run-time Engine Installer** – installs prerequisite software on end user device to run FlexApp One Applications. See the **Prerequisites to Run FlexApp One Applications** section below.

FlexApp One Bundler

The FlexApp One bundler requires the following for operation:

Component	Requirements
OS Platforms Supported	Windows 10 or Windows Server 2016/2019. Only 64-bit versions where applicable are supported. Both physical and virtual instances are also supported.
CPU	2 CPUs minimum
Memory	1 GB minimum
Storage Space	Enough storage to hold the resulting FlexApp One executable. The resulting FlexApp One application will be approximately 25% smaller than the initial FlexApp package.

FlexApp One Applications

The FlexApp One engine is required to run FlexApp One Applications on end user devices. The FlexApp One engine requires the following:

Component	Requirements
OS Platforms Supported	Windows 10 or Windows Server 2016/2019. Only 64-bit versions where applicable are supported. Both physical and virtual instances are also supported.
CPU	2 CPUs minimum, 4 CPUs preferred

Component	Requirements
Memory	2 GB minimum, 4 GB preferred
Storage Space	Enough storage to hold each used application, SSD or better preferred
Limits	4 GB compressed EXE

Prerequisites to Run FlexApp One Applications

To run FlexApp One applications, the FlexApp run-time engine must be installed on the end user's machine. The run-time engine can be included in the base image or installed one time from a FlexApp One application. The FlexApp One bundler will include the necessary files in the resulting EXE. To install the run-time engine from a FlexApp One application:

A. Run any FlexApp One EXE to install the FlexApp service and engine. This requires privilege elevation.

OR

B. Run the Installer.exe using the **install** command line argument. Please note all command line arguments use two hyphens.
   - `Install.exe --install`

Well-Managed Devices

FlexApp One is designed to playback applications in a user's environment from a self-contained executable. These applications run as if they are natively installed in the user's Windows environment without modifying the base image. Windows system files and the Registry remain unchanged keeping the base image clean unlike what would happen when a traditional application is installed or uninstalled. Having a clean, well-managed device provides a repeatable, consistent playback for each FlexApp One application. However, in order to maintain that experience, these end user devices need to start out as clean, well-managed devices. Playback of FlexApp One executables on devices that have been modified with the addition of other software may give unexpected results. The following are recommended:

- End-user devices should have a base image installed without additional applications.
- End-users should not be allowed to install non-managed software.
- Packaging and bundling of software should follow packaging best practices and occur on virtual machines that can be rolled back to a clean state after each run.

Licensing Considerations

FlexApp One is available to customers who have purchased ProfileUnity with FlexApp or FlexApp Only licenses. To request your FlexApp One license, please email your Liquidware Sales Representative or contact us. Customers current on support can continue to create FlexApp One applications from FlexApp packages. When support expires, the FlexApp One bundler can only be used on versions of the bundler that were released before the support

expiration date. Existing FlexApp One applications will stop working when evaluation trials and subscription contracts end but will continue to run for perpetual software customers.

About FlexApp One Applications

Naming

Make the final FlexApp One application a simple static name. The user's registry will be instructed to start an application with a specific path (including filename) when you use `--startup` or 'Activate at Login'. The user's system tray will show the original FlexApp package given name before it was bundled as a FlexApp One application. This can help with version identification.

Storing

FlexApp One applications should reside somewhere permanently since each EXE contains the application read-only data bits. Do not delete the FlexApp One EXE after you launch it. Put your FlexApp One EXE on the fastest storage. It is recommended you store your EXE on local storage which allows for faster and more reliable operation. If your EXE resides on a remote network location, you may be subject to SMB file handle reconnect issues should your connection drop for any reason.

Cloud Syncing

If using Cloud storage for FlexApp One applications, complete the following steps to update executables:

1. Copy the updated application to cloud storage using a different name.
2. Rename the current, in-use EXE to *.old.
3. Rename the new updated application to the simple name of the now .old EXE.

These operations can be staggered to allow endpoints to sync down. When the user logs out or reboots, any binaries in use will close and the renames can occur.

You can also use the "--replace" argument to replace the application if space is a concern, but the client may have to wait for the binary to completely sync down prior to seeing changes to a large application. Rebooting can ensure processes are closed for cloud storage to process changes.

Sample Use Case

For our example use case, we have an organization with laptop users who are frequently offline or working outside of the corporate network. These users need to be able to run applications no matter where they are or whether they are online or offline. FlexApp One allows us to create modular applications that can be delivered using your preferred software distribution system. Successful deployment of FlexApp One applications depends on adherence to application packaging best practices. End user devices should be clean machines with a base image loaded. FlexApp One applications should not replace previously installed software. Previously installed software can leave behind settings that may interfere with the operation of a subsequent FlexApp One application.

FlexApp One Workflow

Here is the main workflow using FlexApp One:

1. Create a traditional FlexApp package.

2. Bundle an existing FlexApp package into a FlexApp One application.
3. Deliver the bundled FlexApp One executable on your favorite software distribution system.
4. End users can double click the shortcuts provided in the FlexApp package (preferred) by using the --CTL option, or the FlexApp One application executables to run the software.

Creating a FlexApp Package

Creating FlexApp packages is done with the FlexApp Packaging Console. The FlexApp Packaging Console is well integrated with ProfileUnity for optimal application delivery. The FlexApp Packaging Console is available for download from both the ProfileUnity Management Console and from the FlexApp One downloadable ZIP file. The installer can be found in the FlexApp Packaging Console folder of the FlexApp One ZIP file.

For more on information on packaging, see the *FlexApp Packaging Console Manual*. Additionally, review these best practices on how to optimize the FlexApp Packaging Console for the best results.

When run with the ProfileUnity Console, the FlexApp Packaging Console will authenticate with the ProfileUnity Service. If you will not be running ProfileUnity, you will need to create your FlexApp packages offline. See this KB article for more information on logging in to the FlexApp Packaging Console in offline mode.

After the FlexApp packages have been created, they can be bundled into FlexApp One executables.

Setting up the FlexApp One Bundler

The bundler is required to create an application executable from a traditional FlexApp package.

1. Download the FlexApp One ZIP file from **Tool & Resources** for the ProfileUnity documentation.
2. Unzip the file.
3. Copy the **Bundler** folder and all its contents to a location of your choice. The bundler can be run from a folder on your FlexApp Packaging Console or another Windows machine.
4. Copy the FlexApp One license that was emailed to you into the same directory as **bundler.exe**.

Creating a FlexApp One Application from a FlexApp Package

FlexApp One applications start out as traditional FlexApp packages which are then bundled with the FlexApp run-time engine to become distributable executable files. To bundle an existing FlexApp package into a FlexApp One application, complete the following instructions:

1. Start an elevated Windows Command Prompt.
    a. Search for Windows Command Prompt and right click the icon or app name.
    b. Select **Run as administrator**.
2. Change the directory to where you copied the FlexApp One Bundler.
3. Start PowerShell by typing `Powershell` and hit the **Enter** key.
4. To allow the FlexApp One Bundler to operate silently, set Bypass on the execution policy. In the PowerShell console, type the following and hit the **Enter** key:
    - `Set-ExecutionPolicy Bypass`
5. Use Windows Explorer to browse to the folder where the FlexApp package is stored. Copy the path of this package folder to memory.

6. Run the FlexApp One Bundler from the PowerShell command line using the path you copied as the argument leading to the FlexApp package. Use quotes around the argument path to the FlexApp package to bundle.
   - `.\Build-engine.ps1 "\\Path\To\The\FlexApp\Folder"`

Sample Bundler Command Line Usage

You can run the Bundle.exe with command line options or run the Build-engine.ps1 and specify the FlexApp application folder to bundle into an EXE. The Bundler.exe command line example and additional options are shown below.

Use the PowerShell script like below:

```
.\Build-engine.ps1 "\\Path\To\The\FlexApp\Folder"
```

OR

```
\bundler.exe bundle -p "\\Path\To\The\FlexApp\Folder" -f
"X:\FA1Bundler\Prod\flexapp.zip" -o "\\Path\To\The\FlexApp\Folder" -t
"X:\FA1Bundler\Prod\engine.exe" -seticon -s 262114 -l
"X:\FA1Bundler\Prod\bundle.log"
```

Additional Bundler Options

FlexApp One Bundler options require only a single dash. You can use the single letter or the whole word as the option.

Option	Description
-OUTPUT	Use -o to steer the resulting EXE. Or in the .PS1, set -o to the "Capture Path" so the EXE will appear in the original FlexApp path where the application was captured to.
-EXPIRATION	Use -e yyyy-mm-dd to set an expiration date for the .EXE. *Example:* To set expiration date for the EXE for the bundler command within the .PS1: `-e 2021-06-30`
-LICENSE	Bundler will give a temporary, 15-day trial to each EXE. For a non-trial EXE, you must have (in the bundler folder) a "FlexApp_One.lic" file with the license string inside of it.
-MIGRATE	The migrate process will migrate new license numbers and support dates. • Updating a self-contained FlexApp One EXE • Updating the engine within your portable FlexApp.exe Use migrate to upgrade the engine within your existing FlexApp executables without re-bundling, even without access to the original package. This process leaves the original package in place in "*.exe.old" form.

Option	Description
	Example: `.\bundler.exe migrate -p Z:\fa1\folder\YourFlexApp.exe -t c:\bundler\prod\engine.exe -l c:\temp\bundle.log`
-SETICON	Will set the output file to the package default icon.
-ICONFILE <path\file.ico>	When used with -seticon, -ICONFILE will allow you to specify the icon file that will be used when setting the icon for the output file.

FlexApp One Application Command Line Arguments

The following describes the available command line arguments for FlexApp One applications. In some command line arguments below, bracketed placeholders indicate information to be supplied based on your use case. Neither the actual characters such as < and > should be typed in on the command line. This merely represents the information that should be inserted at that point in the command line.

Note: Use TWO hyphens with all command line arguments.

Argument	Description
--admin	Shows additional options in the system try icon for this FlexApp One application.
--install	Installs the FlexApp engine (LWLContainerService) as an elevated process.
--upgrade	Will upgrade older FlexApp service only and will work the same as the –install option if the FlexApp service does not exist.
--uninstall	Removes the FlexApp container service as an elevated process.
--startup	The FlexApp One application will be added to the HKCU\Run with -CTL. (This should be your default method.) Or you can use the "--index 999" option to layer, but not to launch the main shortcut.
--index <#>	Activates this layer and starts the EXE number (0, 1, etc.) in tray order.
--index 999	Activates this layer without starting the main shortcut.
--ctl	Activates Click-to-Layer (CTL) shortcuts only – not the layer.
--addtostart	Adds the FlexApp One application shortcut under a FlexApp program group in the Start Menu. This option provides an additional way to always access the application. You can pin this shortcut as it accesses the binary, activates the application, and passes through to the application if it is already activated.
--stop	Closes handler tray binary process and stops the application layer.
--replace <path to old app>	Replaces and restarts the application if it was running. Specify the new application and then the old application. The new application will have the old application name when finished as this flag REPLACES the previous EXE. Use simple names as your final application name that the user will run. Continue to use versions in the package name when you package. The tray will show the package's given name which can identify the version running in the workspace. Example: NewUpdated.exe --stop --replace C:\LocalCopy\OldApp.exe

Argument	Description
--reg	Creates a registration key at: HKCU\Software\FlexApp\App.exe\%username% This can be used in a variety of ways, for example, as a user Microsoft Endpoint Manager detection rule.
--clean	Removes the shortcuts and registry key at: HKCU\Software\FlexApp\App.exe **Must use with --stop.**
--sync <path>	Block level copy of the file to an existing folder. Example: App.exe --sync C:\FlexApps\App.exe
--system	Creates shortcuts in the Public areas versus User areas and will create registry tracker in HKLM if using the "--reg" option.
--remove	Removes the FlexApp One application. Use with "--stop", "--clean", and often "--system" options.
--skipactivation	Will carry out the runtime commands, but not launch the application.
--extract <path>	Creates a VHDX, ICO, XML which can be imported into the FlexApp Packaging Console and edited
--debug	Runs the engine in debug mode. Can be used with –console.
--console	Runs the engine outputting the log to the console.
--persist	Used to persist changes to the application. Typically used when users are local administrators and self-update or extend their applications.

Microsoft Endpoint Manager Deployments

FlexApp One applications can be distributed to end users using Microsoft Endpoint Manager. This application distribution process has two stages. Below is an overview of the process with detailed steps to follow.

1. Deploying the FlexApp Service to the end user machines.
2. Deploying FlexApp One applications to run on end user machines.

Process Outline

The following outlines an example process.

1. Upload your StagingFlexApp.exe to Intune and deploy it with the System context. This will be the FlexApp that will prepare your workspace. This can be a hello world application which just

installs the FlexApp service, or this can be a more complete application which installs the run-times and other prerequisites needed for the base OS. (`--install` or `--upgrade`)

2. Upload and deploy each FlexApp One application with the `--system` command line argument which will distribute the application to the endpoint.

   - `--sync C:\ProgramData\FlexApp\App\FlexAppOne.exe --system --addtostart --ctl`

Detection Rules

1. The FlexApp service detection rule should be based on the presence of a file:

   - `C:\Program Files\ProfileUnity\FlexApp\ContainerService\x64\VirtFsService.exe`

2. The System staged FlexApp Ones will be detected by the presence of a file at:

   - `C:\ProgramData\FlexApp\App\FlexAppOne.exe`

Deploying the FlexApp One Service to End User Machines

1. Install the FlexApp One service on the machine you are delivering the application to. You can install the service from any FlexApp One file using the `--install` switch. Also included is an "installer.exe" which is signed by Liquidware that can be used to deliver the FlexApp One service. If you choose to go this route, convert the executable into an .intunewin file and continue to the next steps.

2. In Microsoft Endpoint Manager Admin Center, click on **Apps** then **All Apps** and choose **Add**.

3. Select **Windows app (Win-32)** from the list that opens on the right side of the screen.

4. Choose the .intunewin file that you converted your FlexApp One file into. Click **OK** to continue.

5. Name the application and provide the publisher. You can choose to fill in any of the other options, but they are not required.

6. On the next screen, we need to tell the service to install. Run the **Installer.exe** with the following settings:

   a. **Install command:** *Installer.exe --install*
   b. **Uninstall command:** *Installer.exe --uninstall*
   c. **Install behavior:** *System*
   d. **Device restart behavior:** *No specific action*

7. Next, choose the requirements. It's common to choose **64-bit** architecture, and any OS newer than or equal to **Windows 10 1703**.

8. Now set a detection rule so that Microsoft Endpoint Manager knows the service has been installed. It is suggested to look for the service executable. Select **Manually configure detection rules** and set the following options:

a. **Rule type:** *File*
b. **Path:** *C:\Program Files\ProfileUnity\Flexapp\ContainerService\x64*
c. **File or folder:** *VirtFsService.exe*
d. **Detection method:** *File or folder exists*

9. The next screen will ask who you want to assign the app to. You can assign it to users or groups as either a required app or an available app, or you can choose to do an uninstall.

10. The final screen is just a review of the previous steps. Double check the install commands and behavior before hitting **Create**. The app will then begin uploading to Microsoft Endpoint Manager.

11. Move on the next section to upload a FlexApp One application and tie it to the service dependency we just created.

Deploying FlexApp One Applications to End User Machines

1. At this point, locate your FlexApp application and turn it into an .intunewin file that you can upload to Microsoft Intune Manger. You can edit your workflow to automatically create an .intunewin file after your FlexApp One application has been created. Within Endpoint Manger, select **Apps** then **All apps** and click **Add**.

2. Select **Windows app (Win-32)** from the list that opens on the right side of the screen.

3. Choose the .intunewin file of the application you wish to upload. For this example, we will use Chrome.

4. The next screen you will need to fill in the app information. Give the application a recognizable name and a description. For example, choose **Publisher** and enter your organization's name. Version the app and add the app logo if you choose. Select **Next**.

5. We now need to program the application. This is one of the key steps. We purposefully built FlexApp One to work with Microsoft Endpoint Manager without the need for additional steps. To make it work we need to use the following commands. (Replace "Chrome.exe" with the application you are uploading)

    a. **Install command:** `Chrome.exe --sync "C:\ProgramData\FlexAppOne\Chrome.exe" --system --ctl --addtostart`

    The `--sync` command and `--system` command are important here. Choose to sync to ProgramData. You can pick any location you choose. The system command is necessary because it will add the icon to all users start menu and desktop if applicable.

    b. **Uninstall command:** `C:\ProgramData\FlexAppOne\Chrome.exe --system --stop --clean --remove`

    This command needs the `--system` switch to make sure the right locations are cleaned up, the `--stop` command to stop the app from running, and the `--clean` and `--remove` commands remove items from the registry startup and shortcuts added by

`--ctl` or `--addtostart` form the desktop or start menu. Also, be sure you point to the path of the executable you set in the previous step.

c. **Install behavior:** *System (This is required to be System.)*

d. **Device restart behavior:** *No specific action*

6. Configure the requirements for the application. In this example, choose **64-bit** architecture and any operating system equal to are greater than **Windows 10 1703**.

7. Configure your detection rule so that Microsoft Endpoint Manager knows the application has been installed. The easiest way to do this is to look for the FlexApp One executable. Select **Manually configure detection rules** and set the following options:

a. **File type:** *File*

b. **Path:** *C:\ProgramData\FlexAppOne*

Choose the path where you installed your app in step 5.

c. **File or folder:** *Chome.exe*

Choose the name of your executable

d. **Detection method:** *File or folder exists*

You can also choose to look at date modified, or date created

8. This is where we will add in the service dependency that we created in the first section of this document. Add a dependency and choose the **FlexApp Service Installer** you uploaded and set it to auto install.

9. Now we need to assign the application to our users. You can assign to individuals are groups and choose whether it should be required or available. You can also choose to uninstall from this screen.

10. Double check all the settings have been set correctly and choose to **Create** the application. This will upload the app to Microsoft Endpoint Manager.

Microsoft AVD, Citrix, Other Remote Desktop Distribution

To deploy FlexApp One applications in remote desktop environments three steps are typically required. All three steps are required for Microsoft AVD and Citrix XenApp environments.

1. Install the prerequisite FlexApp service and run-time engine. This can be done running one of the following:

a. Run the installer in system context:

- `Installer.exe --install`

OR

b. Run any FlexApp One application (this application may optionally contain prerequisites):

- `FlexAppOne.exe --install`

2. Sync the file to a local folder, activate the layer, and do this at the system level using one of the following:

a. If the app is stored on a UNC, run the FlexApp One in system context with:

- `\\server\share\FlexApp\Appname\App.exe --sync`
  `C:\ProgramData\FlexApp\Appname\App.exe --index 999 --system`

OR

b. If the app is local, run the FlexApp One in system context with:

- `C:\ProgramData\FlexApp\Appname\App.exe --index 999 --system`

3. Finish the app deployment by running the local FlexApp One in system context with the following to add the application to the Start Menu, and create a shortcut in the public area without launching the app using the following:

- `C:\ProgramData\FlexApp\Appname\App.exe --ctl --addtostart --skipactivation --system`

OneDrive Deployments

You can distribute your FlexApp One applications to end users using your method of choice. The following outlines a sample process using OneDrive.

1. Share your FlexApp One application on OneDrive with your distribution group.

2. Run a login script to register the apps in the OneDrive folder:
   --startup --addtostart --ctl

Configure OneDrive GPO Administrative Settings:

1. Locate your OneDrive build number:
   a. Click on the OneDrive icon in your system tray.
   b. Click on **Help & Settings** and select **Settings** from the menu.
   c. Go to the **About** tab and write down your build number.
2. Browse to your OneDrive root folder and correct build number. Example path for source ADMX and ADML files.
   - `C:\Users\xx\AppData\Local\Microsoft\OneDrive\21.052.0314.0001\adm`

For English:

1. Take the OneDrive ADMX and place it in a `PolicyDefinitions` folder in: `\\dc\sysvol\domain\policies`
2. Place the ADMX in the `PolicyDefinitions` folder and the ADML in an `en-US` folder under `PolicyDefinitions`.

a. Make sure the language folder exists (`en-US`) and place the ADML file in there.
b. Make `PolicyDefinitions` and `PolicyDefinitions\en-US` folders if they do not exist.
c. Example path for destination ADMX and ADML files.
   - `\\dc\sysvol\domain.local\policies\PolicyDefinitions`
   - `\\dc\sysvol\domain.local\policies\PolicyDefinitions\en-US`

Configure the OneDrive GPO

1. With your OneDrive GPO, browse to Computer **Configuration > Policies > Administrative Templates > OneDrive.**
2. Get the **Tenant ID** and **Sync Path:** to be used when configuring the synced site library.
   a. Login into your OneDrive, locate the folder to be shared and click **Sync**.
   b. Click **Copy ID** near the bottom of the large OneDrive white information dialog box.
   c. Paste this information into the GPO.

Sample Command Line Usage

Launching and activating FlexApp One applications can be done in many ways using a variety of command line options.

For the best single user experience, instruct each FlexApp to create any provided Desktop and Start Menu Shortcuts for the user, to place a shortcut to itself in the Start Menu FlexApp group and to do this each time the user logs in. The command line options for this are **--CTL --AddToStart --Startup**.

For multi-user systems, run in the system context and call each FlexApp with options **--System --Index 999** to mount the application. You can also use **--Sync** to copy the FlexApp One local and then launch it locally when the sync completes. After the initial launch, call each application again with **--System --CTL --AddToStart --SkipActivation** to create any Desktop and Start Menu Shortcuts, add a shortcut to the executable in the FlexApp Start Menu group and skip activating the application since it was already mounted in the prior step.

> ***FlexApp One applications are non-persistent and revert back each time you restart your desktop, unless you use the --persist command line option to persist the applications virtual disk changes.***

The following are commonly used command line options.

System Context Options

Variations and combinations often used for multiuser workspaces depending on distribution solution or use case:

```
--system --addtostart --index 999
```

```
--sync c:\ProgramData\FlexApp\App\App.exe --system --startup --addtostart --index
99
```

User Context Options

Variations and combinations often used for single user workspaces depending on distribution solution or use case:

```
--startup --addtostart --ctl --skipactivation
```

```
--sync %appdata%\FlexApp\App\App.exe --startup --addtostart --index 999
```

```
--sync %appdata%\FlexApp\App\App.exe --startup --addtostart --ctl
--skipactivation
```

```
--sync %appdata%\FlexApp\App\App.exe --startup --addtostart --index 0
```

Using Login Scripts

Variations and combinations often used for single user workspaces:

A login script can also be used to executes each FlexApp One executable with command line options. The following is a single user system example.

```
'app.exe --startup --addtostart --ctl'
```

This will add the FlexApp binary to the USERS registry RUN key, add a shortcut to the binary to a FlexApp start menu folder, and add Click-to-Layer shortcuts for the applications executable (if provided in the original package) to the Start Menu and Desktop locations.

Add `--system` to add to the MACHINES registry RUN key.

Support Logs

Support log files can be found in the following locations:

%temp%\ProfileUnity

%windir%\Temp\ProfileUnity

%programdata%\Microsoft\IntuneManagementExtension\Logs (for Intune Deployments)

Getting Help

If you have questions or run into issues while running the FlexApp One Bundler or FlexApp One applications, you can contact Liquidware for help if your ProfileUnity or FlexApp support contract is current. 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