ProfileUnity™
with FlexApp™ Technology

Help Manual

Introduction

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

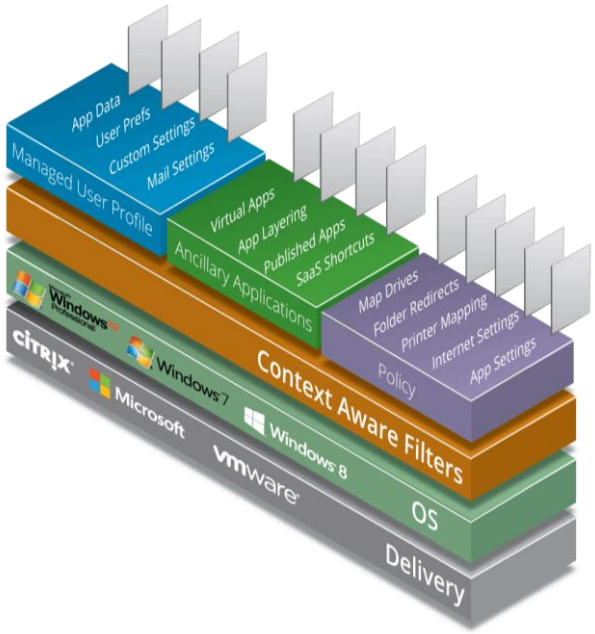
Liquidware Labs, Inc.
3600 Mansell Road
Suite 200
Alpharetta, Georgia 30022
U.S.A.
Phone: 678-397-0450
www.liquidware.com

Contents

ProfileUnity with FlexApp Technology Overview

ProfileUnity™ with FlexApp delivers feature-rich, yet affordable, User Environment Management and Application Layering for both virtual desktop deployments and physical PCs. ProfileUnity decouples user profiles, settings and data from the operating system thus ending the user migration cycle to new Windows desktops, including Citrix XenDesktop, XenApp, and VMware Horizon. ProfileUnity's ongoing User Environment Management features centralize user and policy management with context aware settings that are only limited by your imagination.

ProfileUnity delivers a flexible universal profile that is compatible across multiple Windows versions. ProfileUnity boasts a lightweight agent and no complex software package to install on end-points. This cost-effective solution separately stores and dynamically applies user profiles, configurations, data, and select layered applications to a Windows OS in seconds at login, enabling organizations to be more flexible than ever before with their desktops.



ProfileUnity's FlexApp technology is designed to allow administrators to deploy corporate or department installed applications (DIA) without the burden of tedious base image management. FlexApp enables a potentially limitless number of applications to be stored separately from the Windows operating system yet 'snapped-in' in only seconds at login. Applications look native to the operating system, enabling compatibility of thousands of more applications than with other methods of application virtualization.

ProfileUnity offers these key advantages:

- ProfileUnity retains profiles in native Windows® formats, which keeps them backward and forward compatible across Windows operating systems. ProfileUnity is the perfect solution to harvest user profiles and user data from an older Windows OS and deliver it to a newer Windows OS just in time. Profiles can also co-exist across multiple Windows versions.
- ProfileUnity runs as a standalone system. No proprietary databases are used for user profiles or data. The lightweight client installs to endpoints without the need for software distribution or user downtime. ProfileUnity is priced an average of 40 to 50 percent less than other user virtualization offerings, for faster ROI and a lower overall budget.

- ProfileUnity offers sophisticated features including Context-Aware Filters, Integration with Microsoft® Active Directory, Advanced Folder Redirection options, and more.
- ProfileUnity includes Application Rights Management for no added cost. These features enable you to elevate privileges for select users to run or install select applications and restrict users from running select applications in the base image.

Innovative FlexApp technology provides key benefits to customers, including:

- Simplifies management of non-persistent Windows environments dramatically because fewer master images must be maintained
- Promotes greater desktop virtualization ROI and economies of scale because more workers can use the virtual desktop infrastructure
- Reduces desktop administrator time on maintenance and troubleshooting activities
- Significantly lowers TCO as organizations realize optimal license use and reduce storage requirements
- Layers the application in the native locations throughout the Windows OS without application isolation.

The top reasons that customers deploy ProfileUnity include:

- Speed up overall user login times
- Co-exist Windows Operating Systems – seamless onboarding to new desktop, end user migrations
- Application Layering by group, department, or user – streamline image management
- Application Rights Management – restrict or elevate users to run select apps
- Replace Roaming Profiles – solving profile portability, granular, faster, dependable
- Lower costs of delivering VDI – lower storage and management costs
- Make more users compatible with VDI – knowledge workers and power users can have the customizations and apps they demand even in VDI
- Deliver context aware desktop experience – printer management, settings, shortcuts, etc. all based on custom filters
- Disaster Recovery - persona, data, apps restored in seconds to any Windows desktop
- Ongoing management of the desktop – provision settings, standards, registry modifications, desktop lockdown, and more
- One central user management console -Persona, Applications, Configurations, and central migration settings – for any and all Windows desktops

About the Software

ProfileUnity is now available only as a standalone product and is no longer included as part of Liquidware's Stratusphere virtual appliance. ProfileUnity consists of three parts: the Management Console, the FlexApp Packaging Console, and the Client. The ProfileUnity Management Console provides one central location where administrators can configure persona management and user and machine policies. The FlexApp Packaging Console allows administrators to configure and prepare any applications that will need to be configured for users and made available as a department installed application (DIA). The Client manages each user's settings and persona during their session.

For more information about software requirements and installing the software, please see our ***Installation & Configuration Guide***.

For those planning to utilize cloud storage with ProfileUnity, there is an additional document ***Using Cloud Storage with ProfileUnity*** that you should read. This document guides you through the process of setting up your cloud storage for use with ProfileUnity. The configuration of your cloud storage should be performed before using the ProfileUnity Management Console.

All of these guides can be found on the Liquidware Support Portal along with the rest of our ProfileUnity documentation.

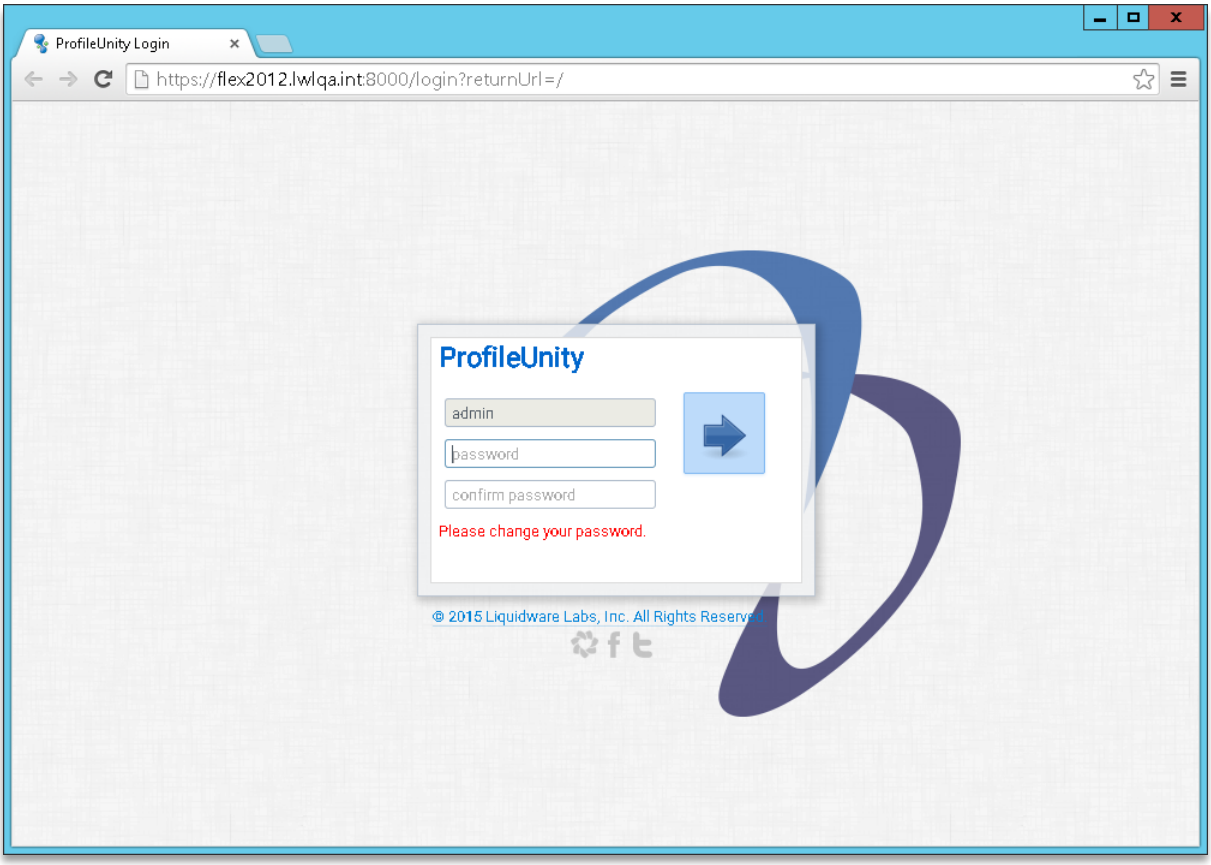
Logging in to the Management Console for the First Time

The first time you log in to the ProfileUnity Management Console, please use “admin” as your username and leave the password field blank. The Management Console will ask you to set up a password for the Admin account at this time.

Your password must be at least 8 characters long and must include at least one character from three of the following categories:

- Upper case letters
- Lower case letters
- Numbers
- Non-alphanumeric characters

Please keep this information in a safe place so you can refer to it as needed.



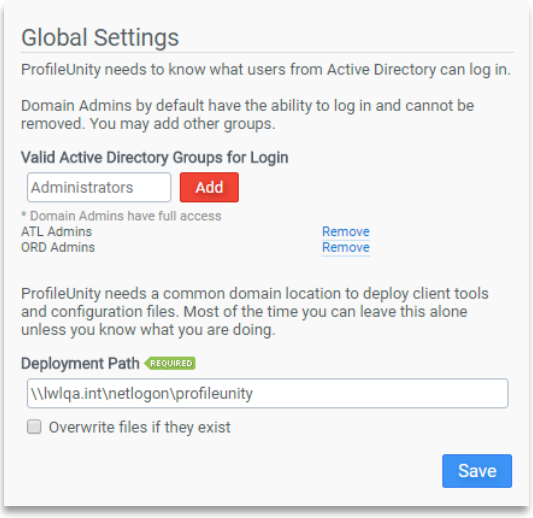
ProfileUnity makes extensive use of Active Directory, including deploying client files and updated configurations to user’s desktops. Consequently, the first time you login to the ProfileUnity Management Console, ProfileUnity will ask you to configure a few global settings to help automate the setup of your environment for ProfileUnity.

You can grant other users access to the Profile Management Console by adding them to the list of valid Active Directory users. Simply type in the AD group and click **Add**. To remove groups, select **Remove** next to that group’s name.

ProfileUnity's client tools and configuration files are typically installed on the NETLOGON share of your domain controller. If you are utilizing cloud storage, you will need point to your "configurations" folder. For example, `S3://lw-profileunity-us-east-1/configurations`. Please read ***Using Cloud Storage with ProfileUnity***, for guidance on configuring your cloud storage. If you need to change where the client files are deployed in your environment, type in the new location for the **Deployment Path**.

These settings can also be edited later in the Administration Area of the Management Console.

Click **Save** to continue.



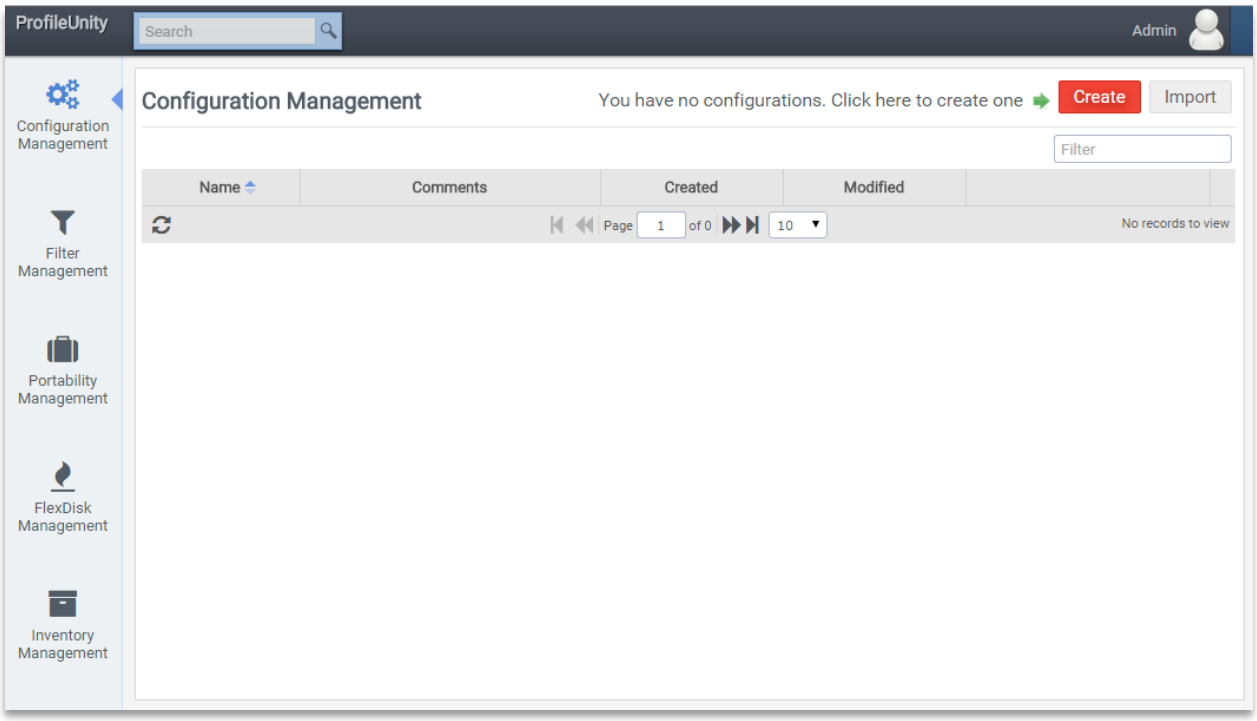
During the configuration process, ProfileUnity offers the option to automatically configure Active Directory using the Computer OU that you setup before installation. However, you must be logged in as an Active Directory admin to the ProfileUnity Management Console for this option to work correctly. Therefore, we recommend that you logout from the ProfileUnity "admin" account and log back in using your Active Directory Domain Admin account credentials before using the Guided Configuration Wizard.

Using the Guided Configuration Wizard

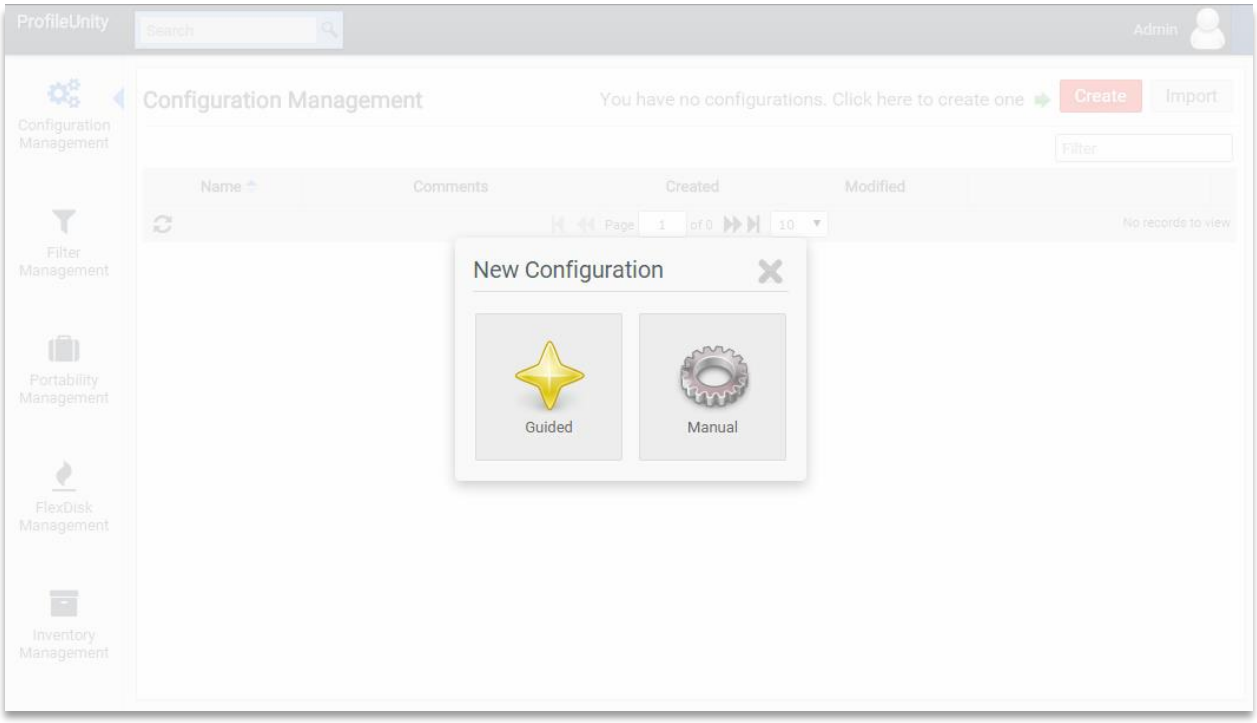
To configure the production environment using the Management Console, ProfileUnity has an easy-to-use Guided Configuration Wizard. The Guided Configuration wizard is a step-by-step wizard that walks you through the creation and assignment of ProfileUnity groups and group policies. It also allows you to easily create a basic configuration for your environment that includes Profile Portability, Folder Redirection and migration of certain folders (if desired), and optionally turns on and configures FlexApp.

The first time you login to the ProfileUnity Management Console, the Guided Configuration Wizard will be started for you to help you automate the setup of your environment for ProfileUnity.

You may also choose to start the Guided Configuration Wizard anytime you create a new configuration from the **Configuration Management** section in the Management Console as seen below.

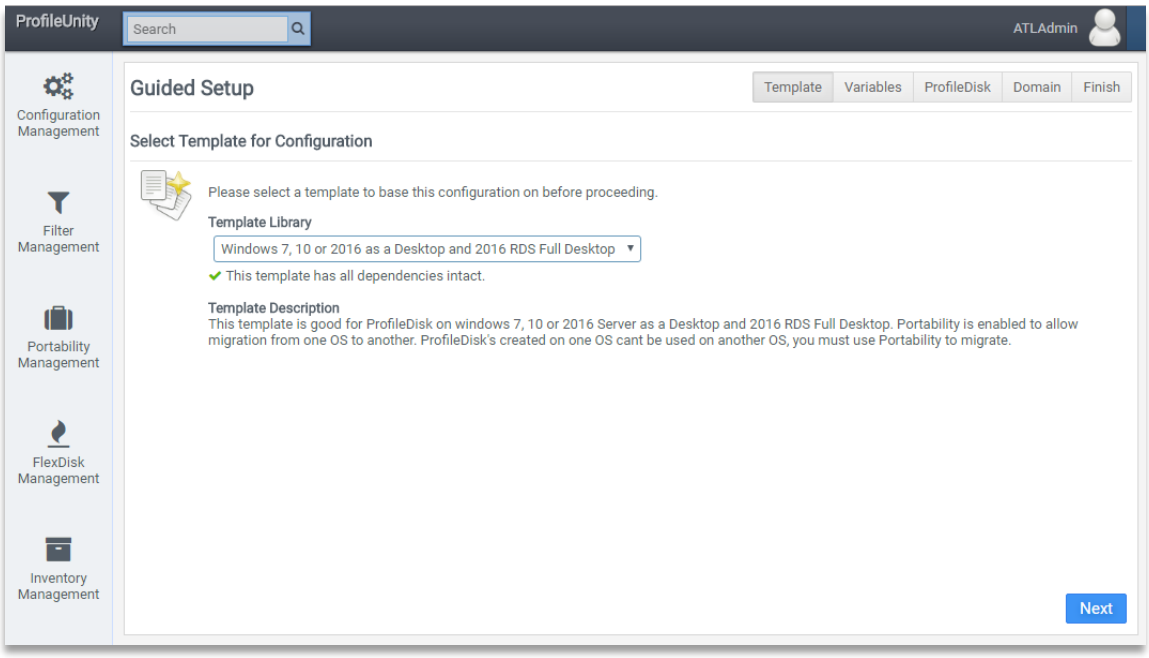


To start the wizard, click on the **Configuration Management** section in the navigation bar on the left. Click on the **Create** button. Select **Guided** to start the wizard for this new configuration.



Select a Template for Configuration

Please choose a template from the library on which to base your configuration. Liquidware has created configuration templates that automatically pre-configure settings based on the goals you are trying to achieve in your environment. You also have the ability to add templates or import templates into this library from the Administration area.



Click **Next** to proceed.

Assign Cloud Storage Credentials

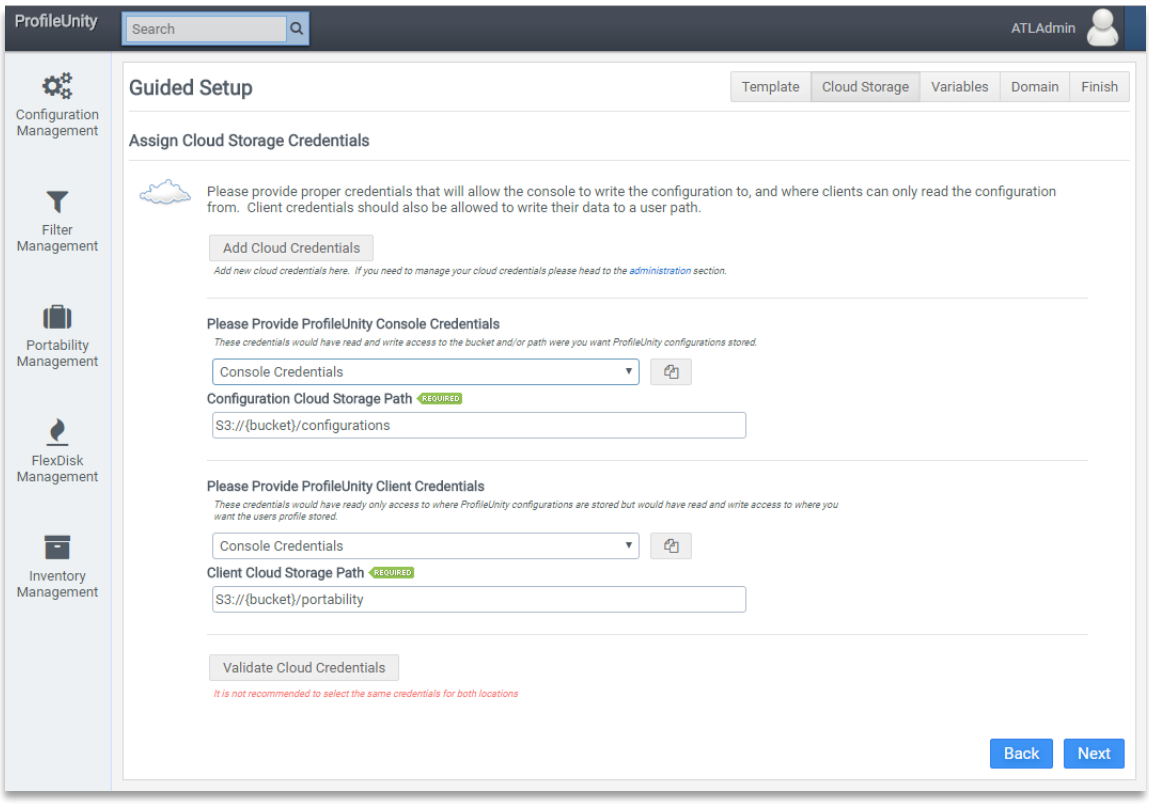
If the template chosen makes use of Cloud Storage for user profiles and ProfileUnity's configuration files, then you will be given the opportunity to configure your Cloud Storage settings. Otherwise, this step will not be included. Please read ***Using Cloud Storage with ProfileUnity*** for guidance on configuring your cloud storage.

If no Cloud Storage credentials have previously been saved, the wizard will ask you for two sets of credentials – the ProfileUnity Console and the ProfileUnity Client user account credentials. You may add other cloud storage credentials by clicking on the **Add Cloud Credentials** button and providing the necessary information. If you need to edit existing credentials, you will need to exit the wizard and go to the Administration area to make any changes in the Cloud Storage Settings section.

In addition to the credentials, you will need to supply the cloud storage path for configuration files, which will need to be in the "configurations" folder. You will also need to supply the cloud storage path for user portability files, which will need to be in the "portability" folder. For example, you may have something similar to:

```
S3://lw-profileunity-us-east-1/configurations
S3://lw-profileunity-us-east-1/portability
```

if you are using Amazon S3 cloud storage.

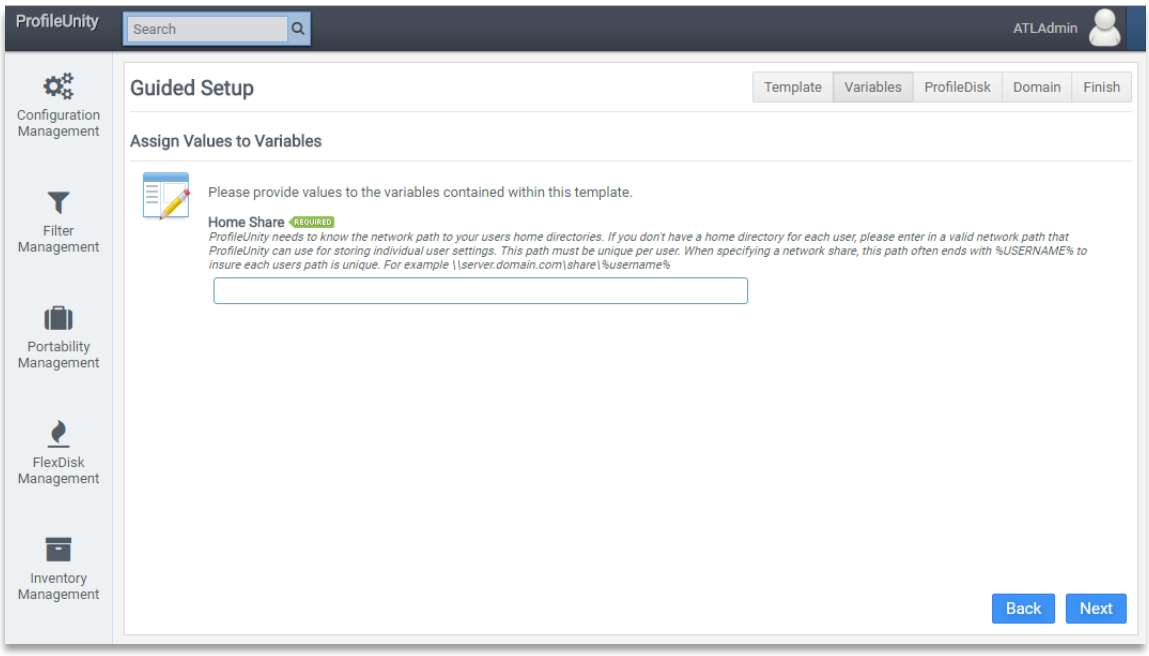


When you are done configuring the settings, click **Validate Cloud Credentials**.

Click **Next** to proceed.

Assign Values to Template Variables

With the use of templates, there will be one or more variables in the template that need values which are specific to your environment assigned to them. For example, you will need to configure a default home directory or location where the user's settings and data will be stored on an existing server in your network. Remember to include `%username%` in the path to make it so that each user will have a unique data folder. Then enter any other values that may be requested.



Click **Next** to proceed.

Storage Path Permissions

ProfileUnity needs the appropriate permissions configured on the storage path for proper operation.

NTFS Permissions

Listed below are the recommended to level NTFS permissions for the storage path.

User Account	Recommended Permissions	Folder
Administrator	Full Control	This folder, Subfolders, and Files
Authenticated User	Modify	This folder only
Creator/Owner	Modify	Subfolders and files only

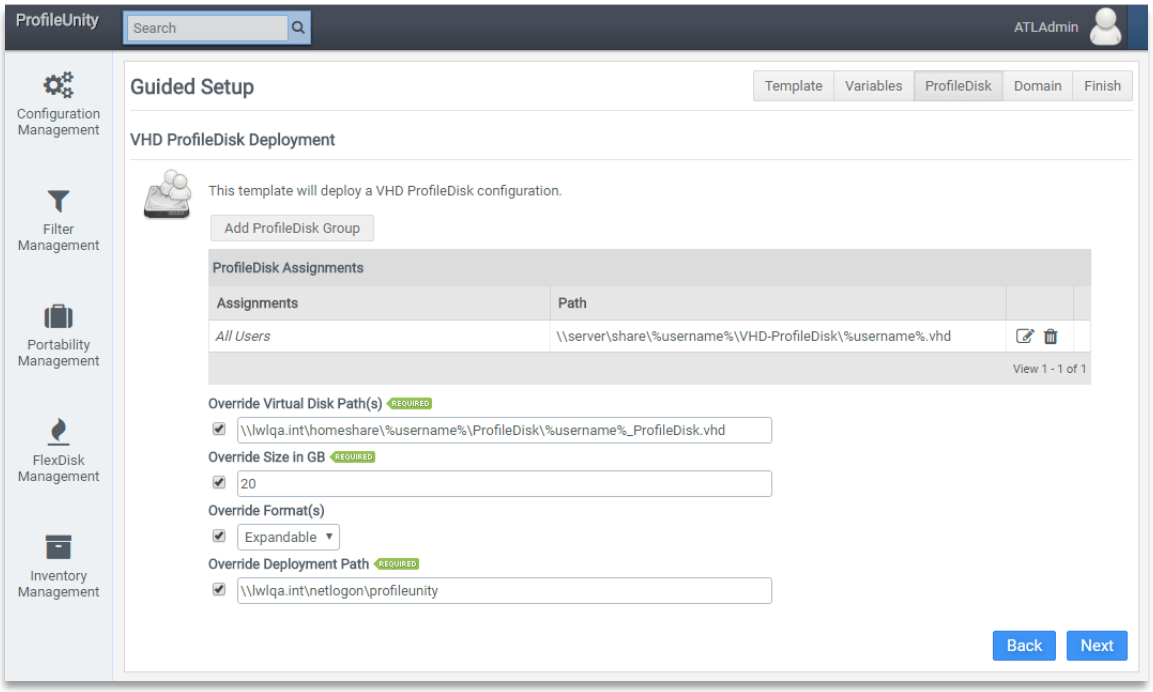
Alternatively, you can specify Everyone Full Control for testing purposes.

Share Permissions

The recommended share permissions for the storage path are Everyone Full Control.

VHD ProfileDisk Deployment

If the template chosen includes the use of a ProfileDisk, then you will be given an opportunity to configure the ProfileDisk settings. Otherwise, this step will not be included.



Click **Next** to proceed.

You will then be asked whether ProfileUnity should deploy the ProfileDisk configuration for you. Select **Yes** for ProfileUnity to automatically deploy the configuration.

Configure Active Directory GPOs

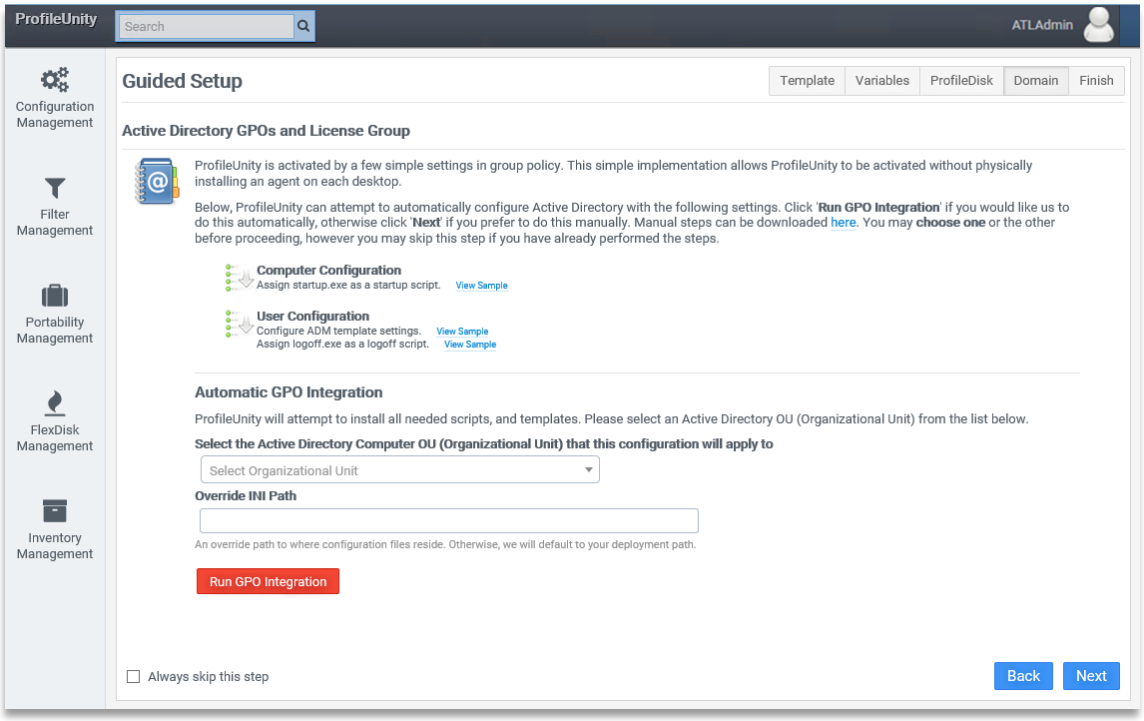
If you are logged in as an Active Directory admin to the ProfileUnity Management Console, the next step in the guided configuration is to configure Active Directory so that ProfileUnity can be activated without physically installing the Client on all of your desktops.

If you click **Run GPO Integration**, ProfileUnity will attempt to configure Active Directory using the Computer OU that you setup before installation. However, you must be logged in as an Active Directory admin to the ProfileUnity Management Console for this option to work correctly.

Otherwise, click **Next** if you prefer to configure Active Directory yourself. Also click **Next**, if you have already configured Active Directory on a prior run through the Guided Configuration Wizard, or if you are not currently logged in as an Active Directory admin. ProfileUnity will ask you to confirm that you do not want to automate the AD configuration. You can always choose to re-login later as an Active Directory admin and have ProfileUnity automate the Active Directory configuration when creating another configuration. Just remember that Active Directory must be configured before ProfileUnity will work for your users.

The instructions for manually configuring Active Directory can be found in the section called **Manually Preparing Your Environment to Use the ProfileUnity Client**.

The AD Configuration is applied globally to all ProfileUnity Configurations. Active Directory does not need to be re-configured for every new Configuration that is created. Therefore, you can choose to skip this step by either clicking on **Next** or you can check **Always skip this step** so that you don't see this step the next time you run the wizard.



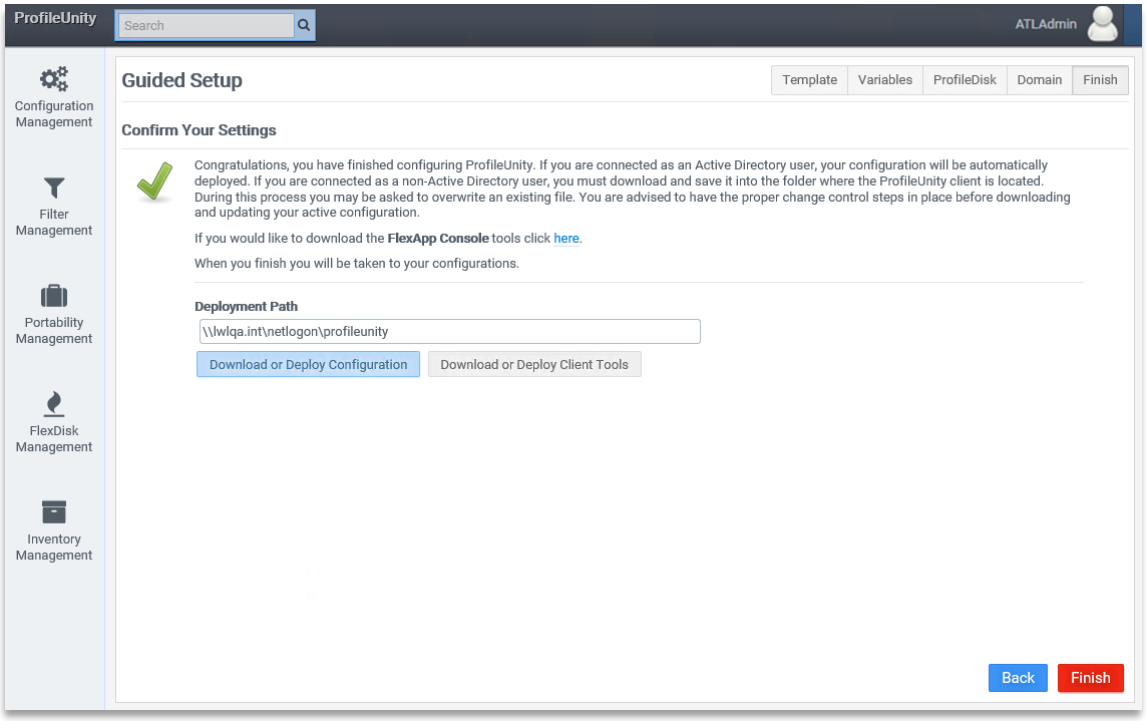
After clicking **Run GPO Integration**, ProfileUnity automatically configures a few items. The wizard will configure the Group Policy Object for the ProfileUnity Computer OU you select. In addition, the wizard will

deploy the client tools to a ProfileUnity subdirectory in the location you specified for the **Deployment Path** in an earlier step. This is typically your NETLOGON share of your domain controller.

When done with the AD Configuration, click **Next** to proceed.

Confirm Your Settings & Save your Configuration

As a final step, the wizard needs to create a configuration that incorporates the settings as you set them in the previous steps and includes basic configuration for all Portability Management rule sets as well. These configuration changes need to be made available to the ProfileUnity Client so that it can apply your settings with each user login. Each configuration is stored in an INI file. Follow the steps below to create this configuration file that will need to be saved to the same location where the ProfileUnity Client is installed.



If you are logged in as an Active Directory user with write permissions to the **Deployment Path**, the configuration file can be deployed directly to the **Deployment Path**. To have ProfileUnity copy the Configuration INI file to the specified **Deployment Path**:

1. Click on the **Download or Deploy Configuration** button.
2. Select the "Domain" **Platform** option and confirm the **Deployment Path** is correct.
3. Click the **Deploy** button.

If you are logged in as a non-Active Directory user or do not have write permissions to the **Deployment Path**, you will need to download and manually copy the configuration file to the same folder where the ProfileUnity Client either is or will be located. To download the configuration file:

1. Click on the **Download or Deploy Configuration** button.
2. Select the "Download" **Platform** option.
3. Click the **Deploy** button.
4. Extract the downloaded ZIP file contents to your **Deployment Path** manually.

For those using cloud storage, please read ***Using Cloud Storage with ProfileUnity***, for guidance on configuring your cloud storage. Your deployment path will need point to your "configurations" folder. For example, your deployment path may look something like this:

```
S3://lw-profileunity-us-east-1/configurations
```

if you are using Amazon S3 cloud storage.

If you are using cloud storage and logged in as an Active Directory user, the configuration file can be deployed directly to the cloud **Deployment Path**. To have ProfileUnity copy the Configuration INI file to the specified **Deployment Path**:

1. Click on the **Download or Deploy Configuration** button.
2. Select the "Cloud" **Platform** option and confirm the **Deployment Path** is correct.
3. Choose your ProfileUnity Console account for the **Cloud Credentials**.
4. Click the **Deploy** button.

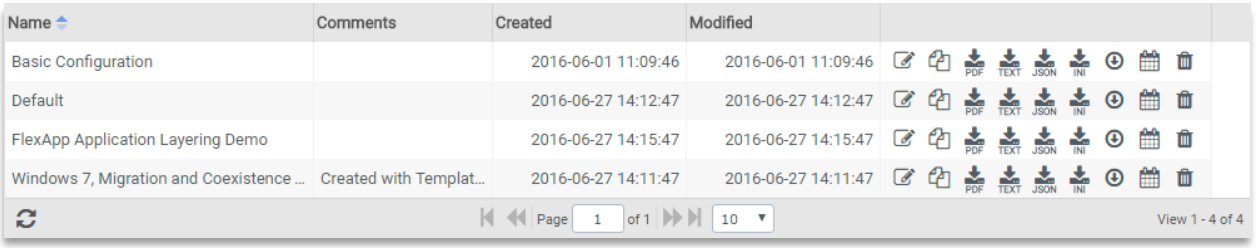
In similar way, you may download or deploy the ProfileUnity Client Tools from this final step of the Guided Configuration Wizard if you haven't done so already. A link to download the FlexApp Packaging Console is also included in this step. Please note that if you choose not to download or deploy either software package now, you can access both the Client Tools and the FlexApp Packaging Console from the Administration section of the ProfileUnity Management Console at any time.

When done with the wizard, click **Finish** to end**.**

Managing Lists in the Management Console

Configurations, filters, and many other objects in the ProfileUnity Management Console are displayed in lists. These list views can be organized and sorted. Let's take a look at some of the functionality before moving deeper into the objects that are available in ProfileUnity.

The lists are displayed in a standard grid fashion with column headings at the top and one item listed per row. The footer row contains some additional options to navigate through the list views.



Working with Columns

The list view columns can be expanded, contracted, or reordered, and the data within each list can be sorted.

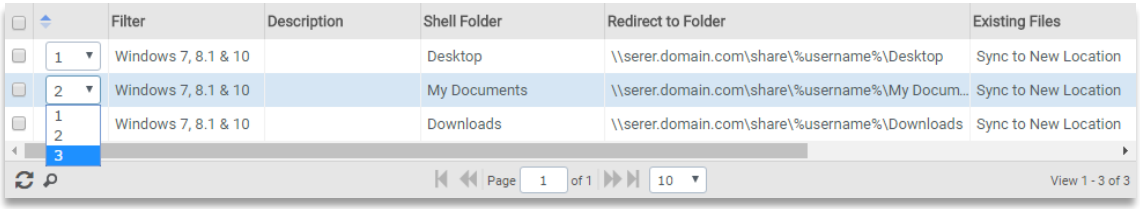
To change a column's width, go to the right-hand border of the column whose width you want to change. Hover your mouse until you see the cursor change to the double-sided arrow. Left-click and drag your mouse right or left until the column is your preferred size.

To change the order in which the columns are displayed, left-click on the column you want to move and simply drag it to its new position.

To sort a list based on a particular column, left-click on the column heading. The column will first be displayed in ascending order. Left-click the column heading a second time to sort the data in descending order. Small up and down arrows will be displayed next the column name indicating which column the list is sorted by and whether it is sorted in ascending or descending order.

Ordering Rows in the List

Some lists can be ordered to indicate in which order a specific rule should be applied. To change the order, left-click on the number column and choose the new order number. Please be aware that other settings in addition to an item's order number may also affect how the item is applied such as when a rule is executed, how often a rule is executed, and whether it is disabled or enabled.

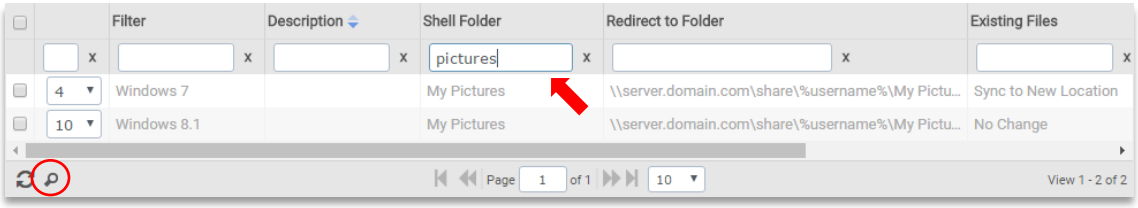


Navigating Using Options in the Footer

There are several more navigational options available in the list view footers. Not all options are available in every list.

To refresh a list view, click on the circular arrows on the bottom left of the list grid. This will reload the grid you are currently viewing.

To narrow down long lists and focus on specific items, toggle the column search toolbar using the magnifying glass icon sometimes seen in the bottom left of the list grid. When the magnifying glass icon is pressed, each column header will include a search bar. Enter text in as many of the search bars as you like to narrow down the number of list items displayed based on the number of matches found. To clear the search criteria for a column, click on the 'x' next to the search bar.

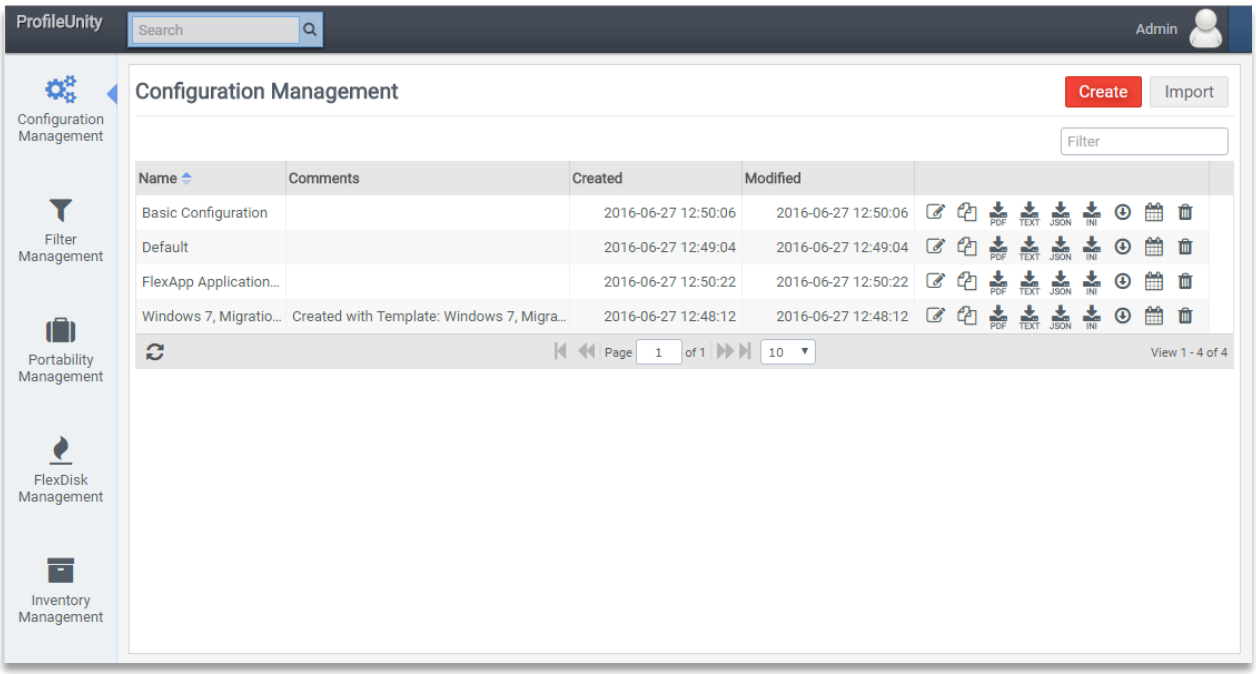


When lists are long, the paging options in the center of the footer will help you navigate. You can select to show 10, 20, 30, 50 or 100 items per list view. The page text box allows you to directly select which page to show in the view. Use the double arrows to one page forward or backward at a time, or use the arrow with a bar to jump to the first or last page of a list.

The numbers on the right of the footer let you know how many total items you have in your list and which of those items you are currently viewing.

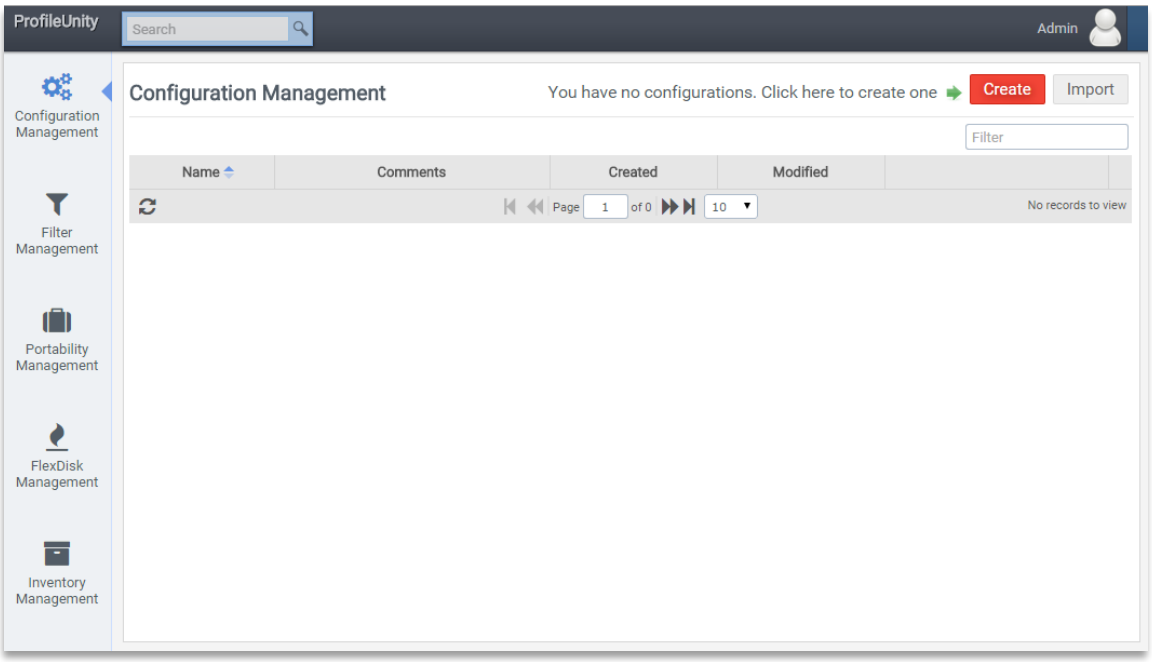
Understanding Configuration Management

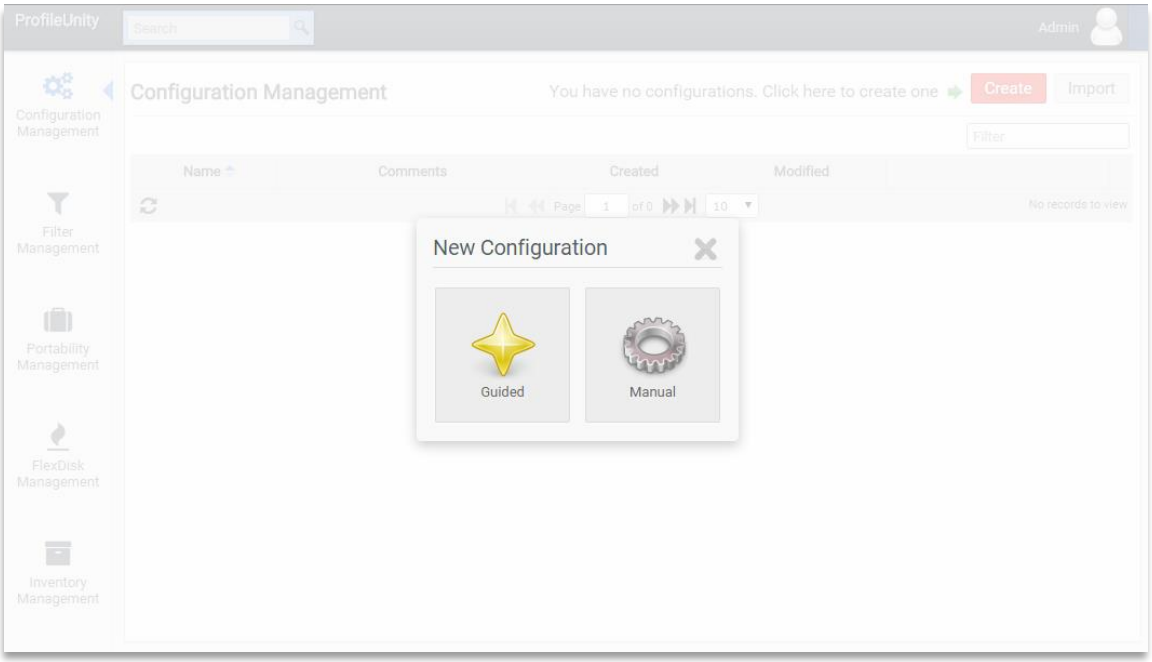
Your user and environment settings are stored in one or more configuration files which are used to control ProfileUnity's execution. During user logon, the ProfileUnity Client reads your configuration files and applies the settings to your client machines. Creating, modifying, deleting, summarizing, and downloading each configuration file is done through the Configuration Management area of the ProfileUnity Management Console.



Creating a New Configuration

If you have not setup any configurations yet, your Configuration Management list will be empty. To create a new configuration for ProfileUnity, you can click on the **Create** button and choose **Guided** or **Manual.** Selecting **Guided** will walk you through the Guided Configuration Wizard and create a basic, editable configuration for ProfileUnity that enables the most popular services including Profile Portability, Folder Redirection and migration of certain folders, and FlexApp. If you select **Manual**, ProfileUnity creates an empty configuration that you can then edit to setup your customizations.


ProfileUnity
Search
Admin
Configuration Management
Filter Management
Portability Management
FlexDisk Management
Inventory Management
Configuration Management
You have no configurations. Click here to create one
Create
Import
Filter
Name
Comments
Created
Modified
Page 1 of 0 10
No records to view


ProfileUnity
Search
Admin
Configuration Management
Filter Management
Portability Management
FlexDisk Management
Inventory Management
Configuration Management
You have no configurations. Click here to create one
Create
Import
Filter
Name
Comments
Created
Modified
Page 1 of 0 10
No records to view
New Configuration
Guided
Manual

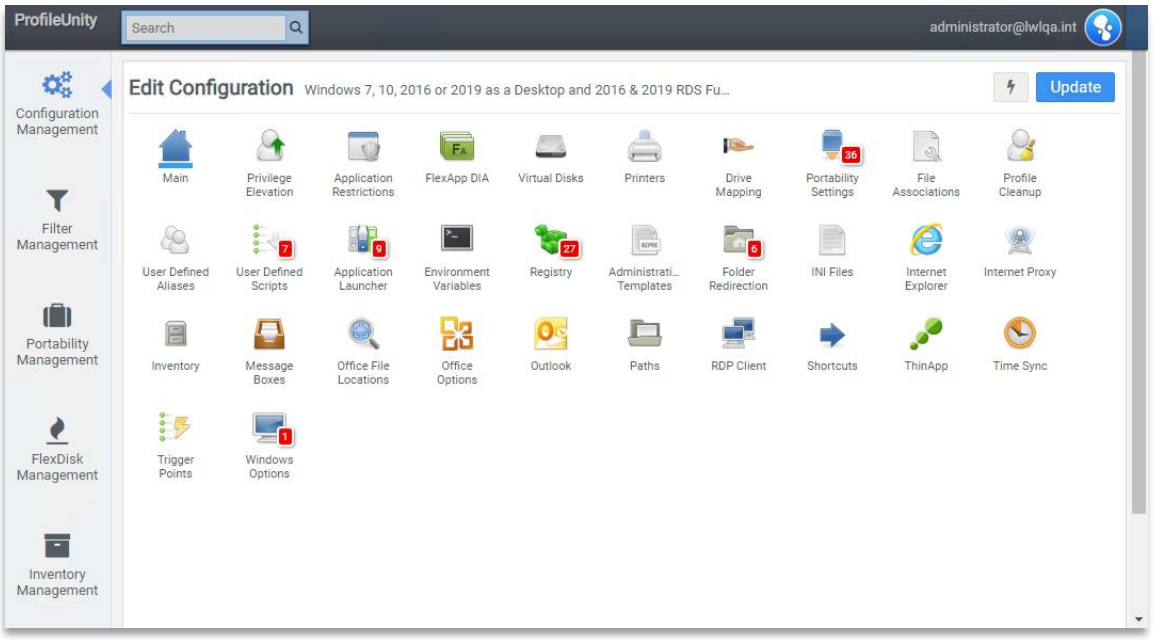
If you have one or more configurations in the Configuration Management list, you may also click on **Copy** next to an existing Configuration's name to clone an existing configuration to use as a starting point for a new configuration. Clicking **Copy** will open the Configuration Editor with a copy of your existing configuration allowing you to rename the configuration and make any other changes to your settings.

Editing Configurations

The Configuration Editor, shown below, is where all configuration settings are entered. To start the Configuration Editor, select one the following options in the Configuration Management list:

Edit – Modifies an existing configuration

Copy – Duplicates an existing configuration



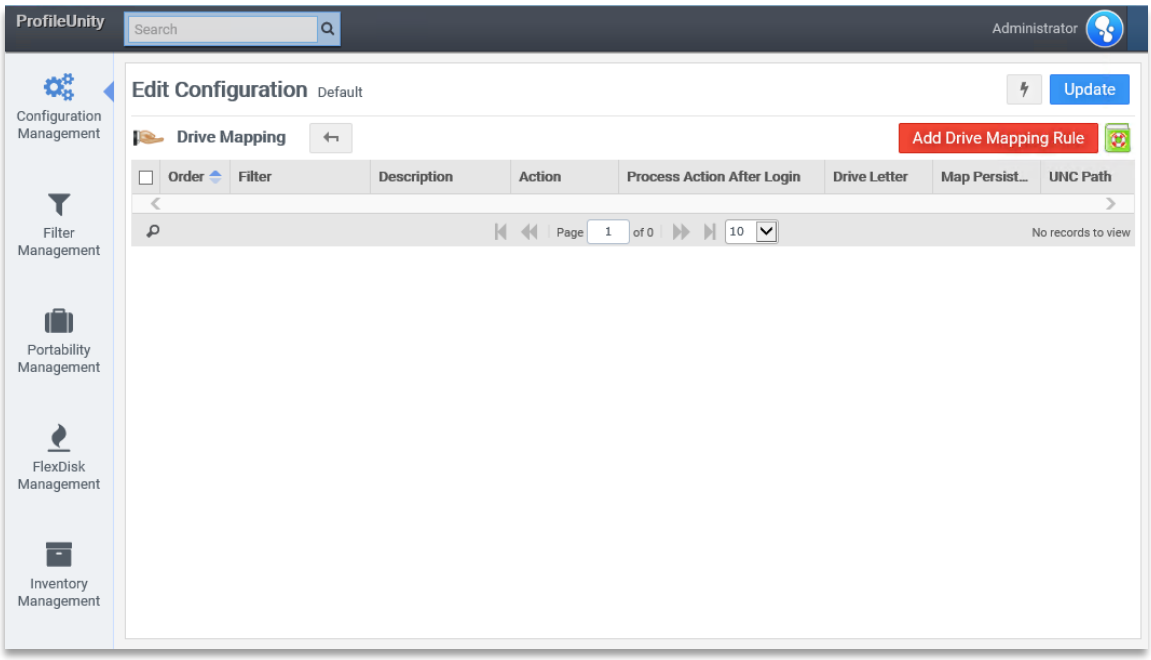
Configuration Editor

A complete configuration is comprised of a group of smaller configuration modules that are designed to complete a specific task. When the editor is invoked, all the available configuration modules are listed. You do not have to configure settings in each of the modules. Customize your settings by choosing the modules that you need to configure for your specific environment. If there are settings within a configuration module, a red box will display next to the name of the module. The number in the red box indicates how many rules have been setup for this particular configuration module.

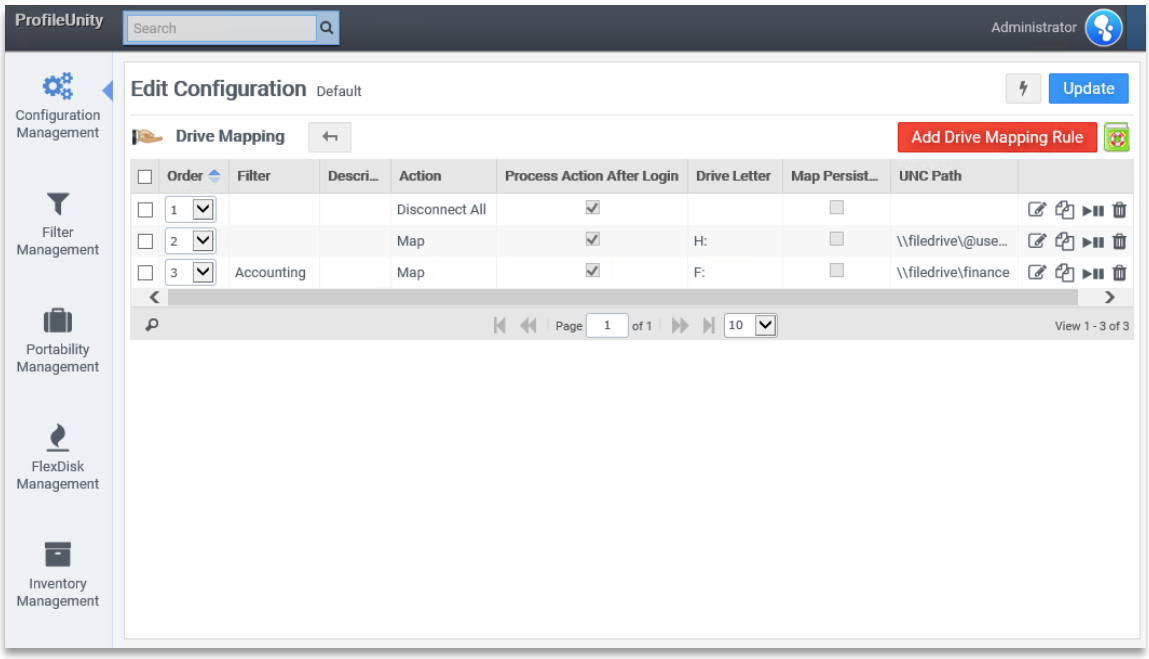
For more information about each individual module, please see "Appendix C - Configuration Modules". In addition, please review "Appendix B – Configuration Macros" to see how to use defined macros in an expression.

Editing Configuration Modules

All configuration modules other than the **Main** module permit you to make multiple entries. Each entry inside a module is referred to as a configuration rule. Selecting any configuration module that permits multiple entries invokes the Configuration Module Editor. The Configuration Module Editor displays the name of the ProfileUnity configuration at the top next to “Edit Configuration” and the name of the specific configuration module that is currently being edited in the left-hand corner. When no configuration elements are defined for a configuration module, the Configuration Module Editor will display as shown.



Selecting the red **Add…Rule** button will create a new configuration element. Each configuration rule you create is displayed in the Configuration Module Editor. Here we see the **Drive Mapping** configuration module with multiple configuration rules defined.



Similar to the icons in the Configuration Editor, the icons to the right of each configuration rule in a module help you manage your settings. To make changes to any rule, click on the appropriate icon:

Edit – Modifies an existing configuration rule

Copy – Duplicates (clones) an existing configuration rule

Enable or Disable – Disables or Enables a configuration rule

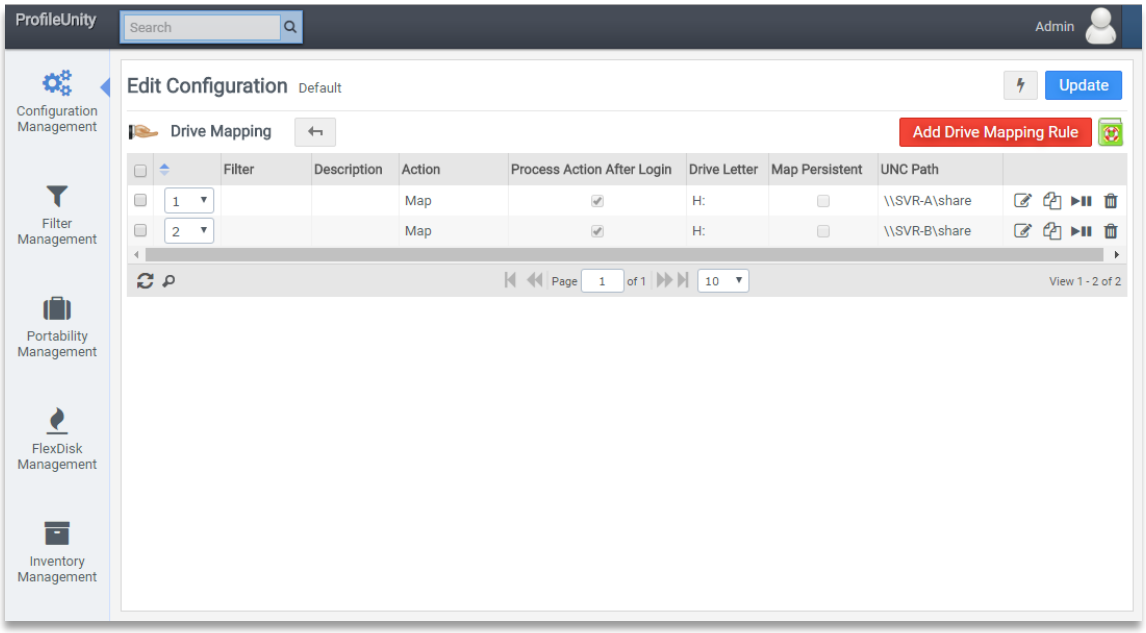
Delete – Removes a configuration rule from the module

Ordering Configuration Rules within the Modules

Configuration rules are processed sequentially starting with the first configuration rule and ending with the last configuration rule. However, the processing order does not indicate which configuration rule takes precedence. Configuration rule order and precedence can best be explained with the following two examples.

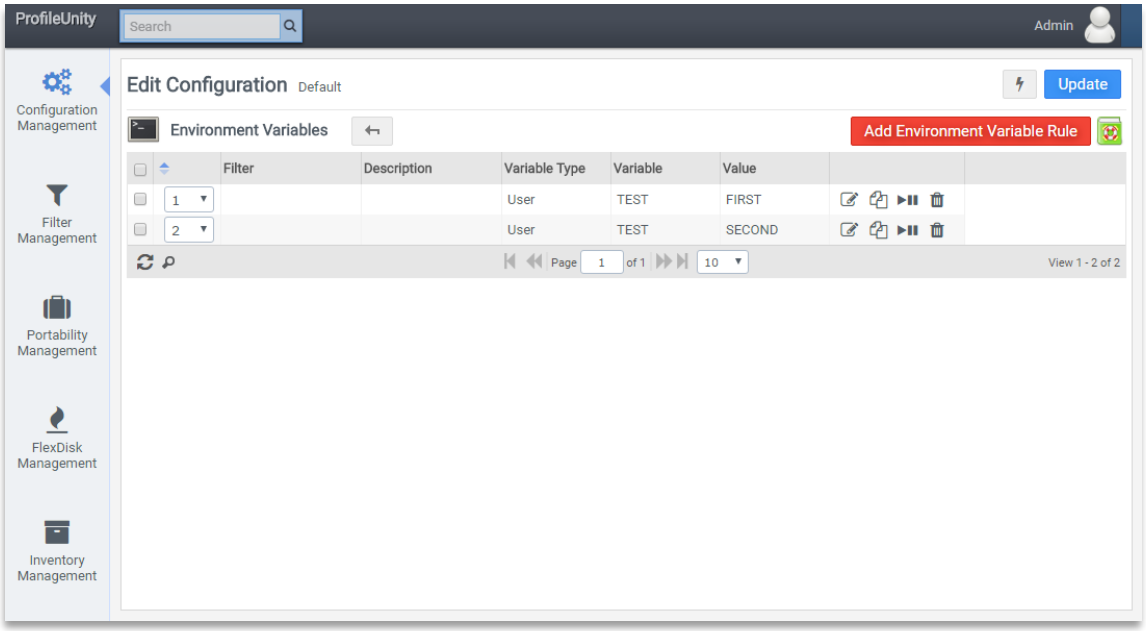
Drive Mapping Order Example:

Here is the **Drive Mapping** configuration module with two configuration rules defined. The first configuration rule maps the H: drive to a share located on SVR-A. The second configuration rule maps the H: drive to a share located on SVR-B. When the first configuration rule is processed, the H: drive is mapped to SVR-A. The second configuration rule will be unable to map the H: drive to SVR-B since the H: drive is already mapped to SVR-A. In this example, the first configuration element processed wins.

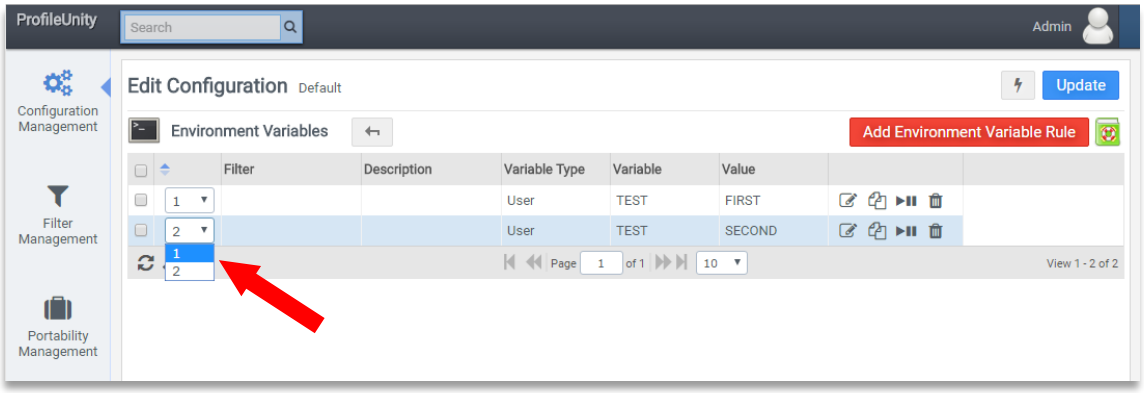


Environment Variables Order Example:

Here is the **Environment Variables** configuration module with two configuration rules defined. Both rules modify the environment variable TEST. When the first configuration rule is processed, the TEST environment variable is set to the value FIRST. When the second configuration rule is processed, the TEST environment variable is set to the value SECOND. In this example, the last configuration rule processed wins.



As you can see from the examples given, the order of configuration rules is important and can greatly impact expected results. You can change the processing order of a configuration rule by left-clicking on the number column and choose the new order number. Please be aware that other settings in addition to an item's order number may also affect how the item is applied such as when a rule is executed, how often a rule is executed, and whether it is disabled or enabled.

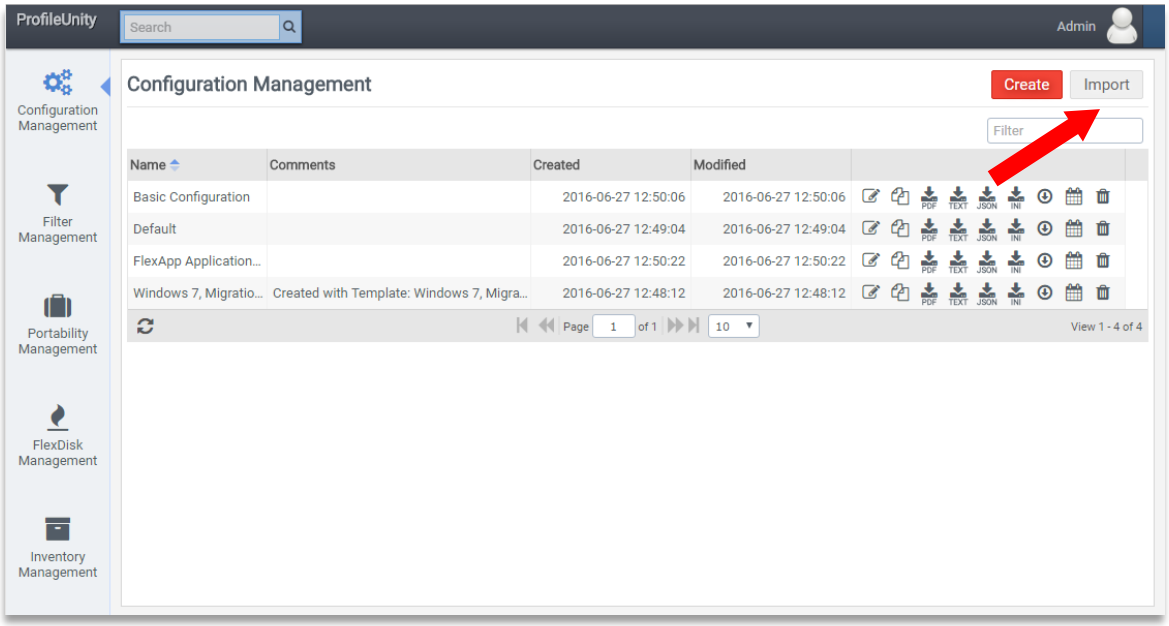


Saving Your Changes

When you have finished editing a configuration, click on the blue **Update** button to save all your changes before leaving the Configuration Management area. If you make changes to a configuration but do not click on the **Update** button before attempting to leave the Configuration Management area, ProfileUnity will prompt you to save your changes. All unsaved changes will be discarded.

Importing and Exporting Configurations

Individual configurations can also be imported into the ProfileUnity Management Console. Just click on the **Import** button at the top right of the Configuration Management area and specify the location of the JavaScript Object Notation (JSON) configuration file. The new configuration will be added to the Configuration Management list. When the configuration is imported, all filters and portability rules that help make up that configuration will be created as part of the process. However, if there are already existing filters or portability rules with the same exact name in the Management Console, the existing settings take precedence and the settings in the JSON configuration file will not be imported.



To export or backup an existing configuration:

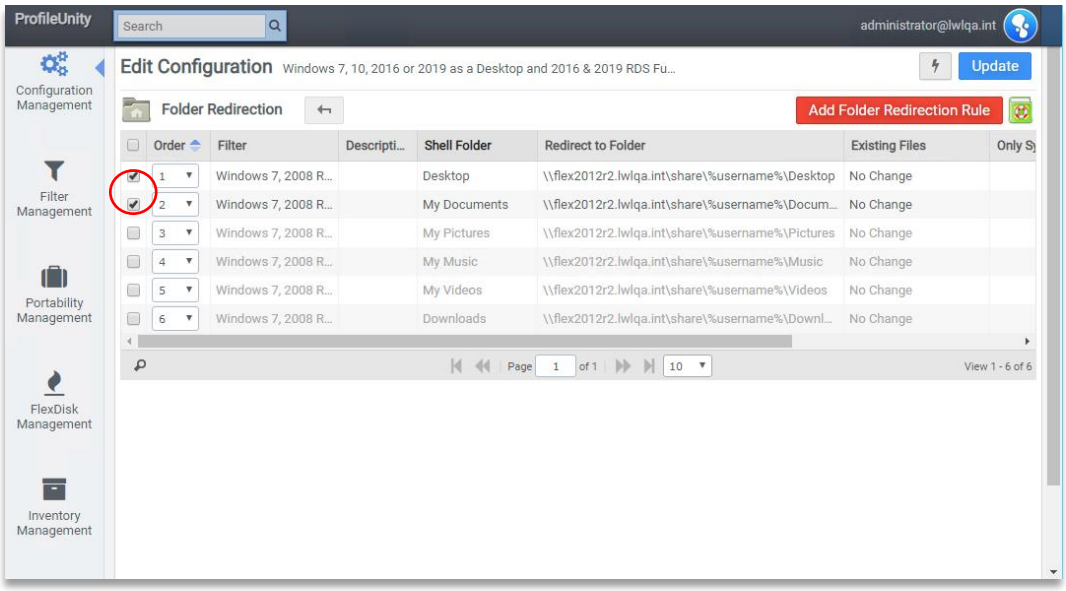
1. Go to the **Configuration Management** area and locate the name of the configuration in the list.
2. Select the **Export Configuration** option from the configuration's row. 

3. Specify the location where the JSON file should be saved.

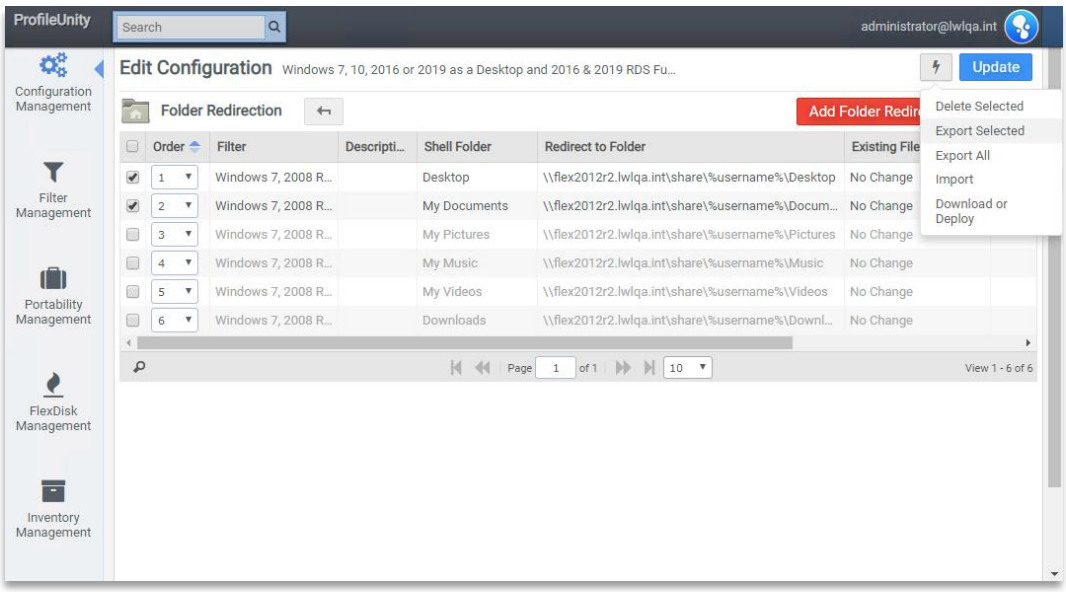
Selective Importing and Exporting of Individual Configuration Module Settings

ProfileUnity also enables you to selectively choose which settings in each Configuration Module to export so they can be shared among other Configurations. One or more settings can be exported from one or more Configuration Modules at the same time. These settings can then be imported into another existing Configuration to supplement the settings that already exist in that Configuration.

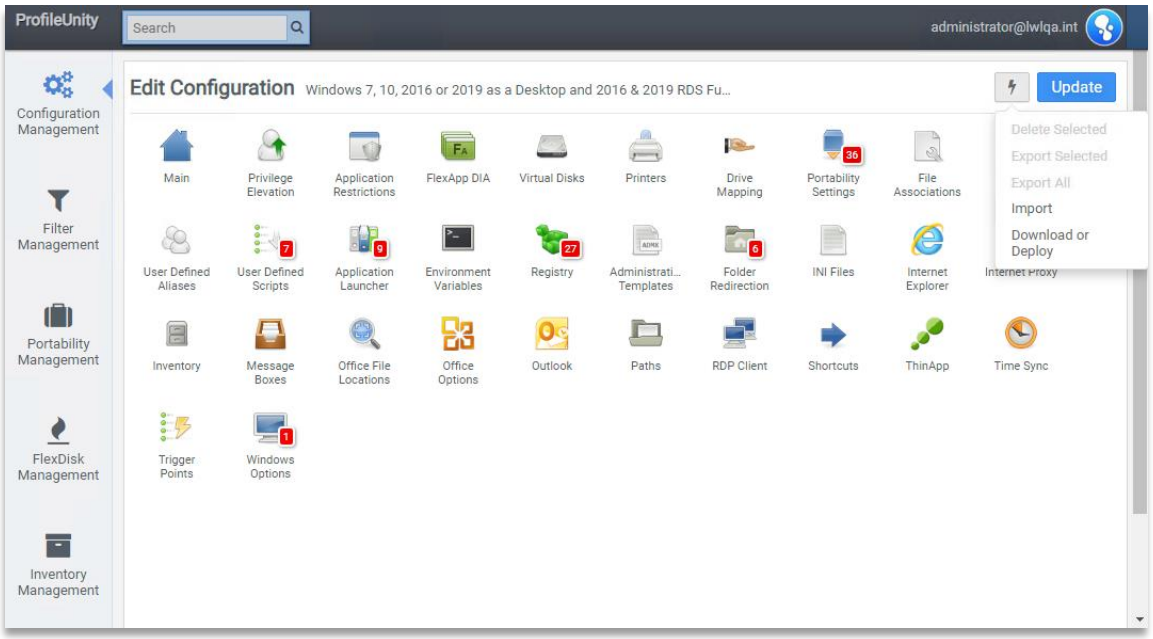
Go inside each Configuration Module and select the particular settings you wish to export.



When you have finished selecting the settings to export, click on the lightning icon in the upper right next to the Update button. Select **Export Selected**. This will download a JSON file.



To import this collection of settings to another Configuration, open the receiving Configuration for editing. Click on the lightning icon, and select **Import**. Each individual setting will be added to its associated Configuration Module in the receiving Configuration.



Please note that while a partial Configuration export can be imported into other existing Configurations, a partial export cannot be imported as a brand-new Configuration since it does not contain all the required settings.

Activating Configurations

Once you finish making changes to your configuration, you need to make those changes available to the ProfileUnity Client so that it can apply your settings with each user login. Each configuration is stored in an INI file which is saved in the same location where the ProfileUnity Client is installed. To create an INI file:

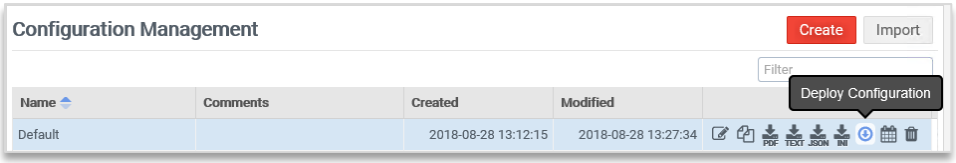
1. Go to the **Configuration Management** area and locate the name of the configuration in the list.
2. Click on the **Download Configuration** icon from the configuration's row. 

3. Save this INI file to the same location where the ProfileUnity Client is installed.

Anytime you make a change to a configuration you must re-download the INI file. In addition, those changes will not take effect until the next logon event from the user. If they are currently logged in when the configuration file is downloaded, they will still be under the old configuration settings.

Automatically Deploying ProfileUnity Configurations

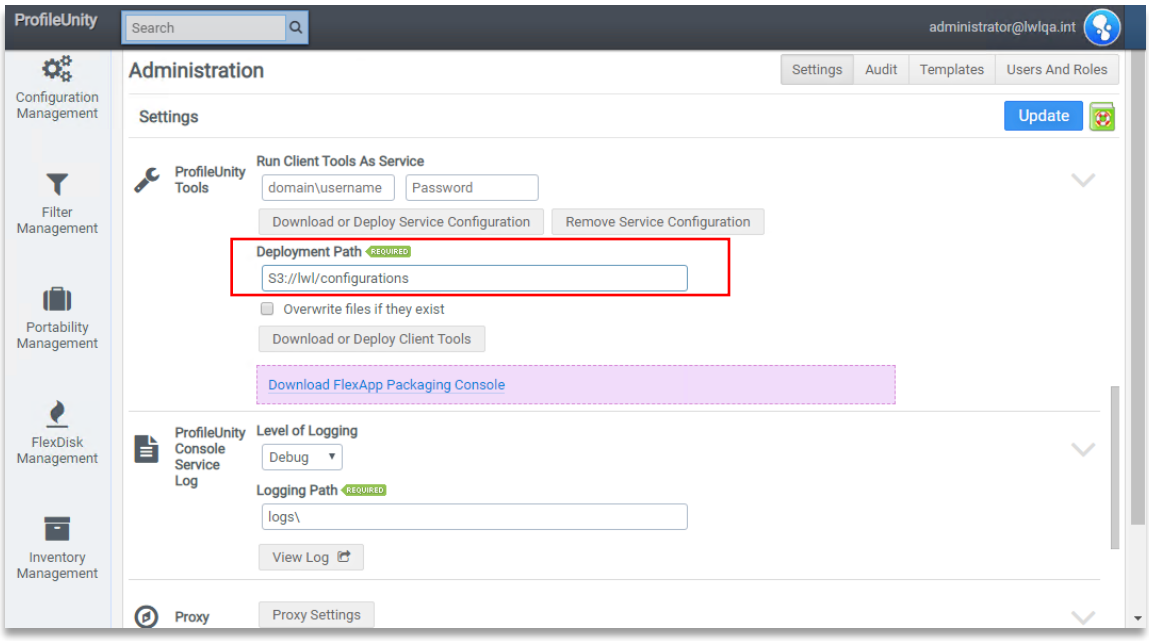
ProfileUnity also has the ability to automatically deploy your configuration changes to a specified location, so that you don't have to manually download and copy the files each time. From the Configuration Management List, click on the **Deploy Configuration** icon to the right of the name of the Configuration.



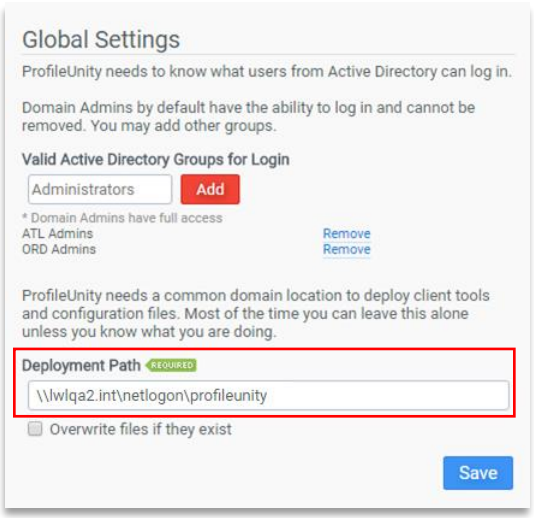
Configuration files are INI files that must be copied to the same location where you installed the ProfileUnity Client, which is known as the Deployment Path. Configuration changes take affect the next time the user logs in. If a user is already logged in when a new configuration file is deployed, their session will still use the old configuration settings until they log out of their current session.

There are several places you can specify or change the Deployment Path in the ProfileUnity Management Console.

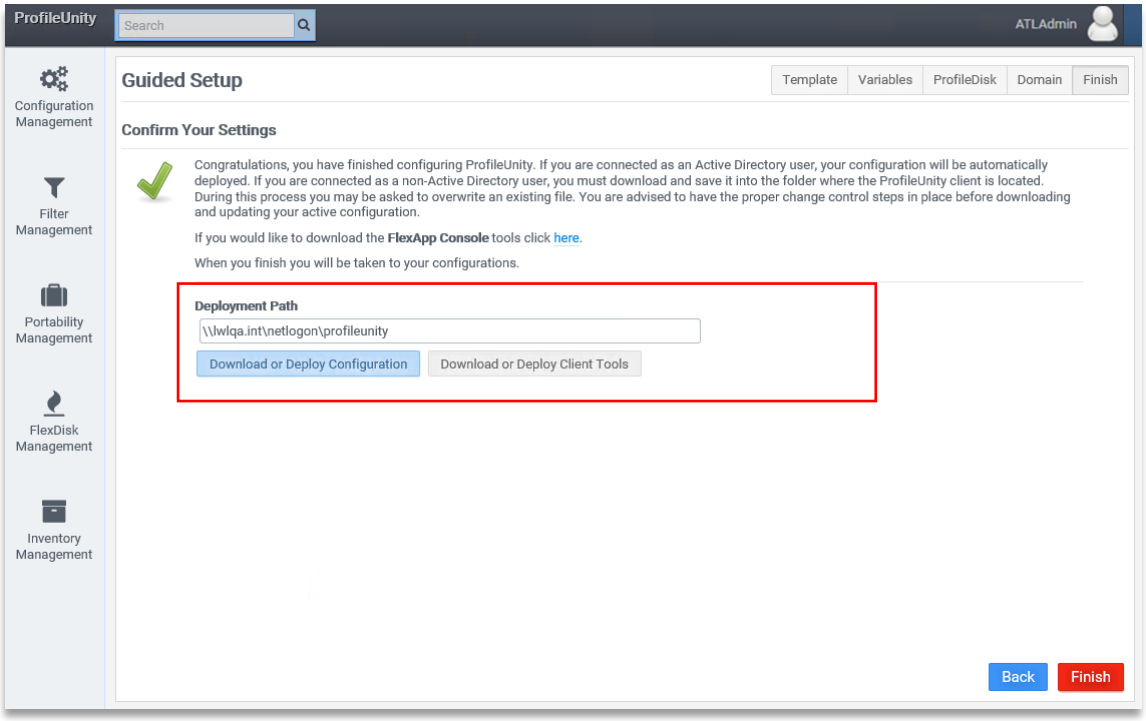
The Deployment Path is stored as part of the Administration settings. Login to the ProfileUnity Management Console and go to the Administration area. To get there, go to your user name at the top right of the Management Console and select **Administration** from the drop-down list. Scroll down to the ProfileUnity Tools section. In the **Deployment Path** field, enter the location where the ProfileUnity Client is installed and click on the blue **Update** button at the top right.



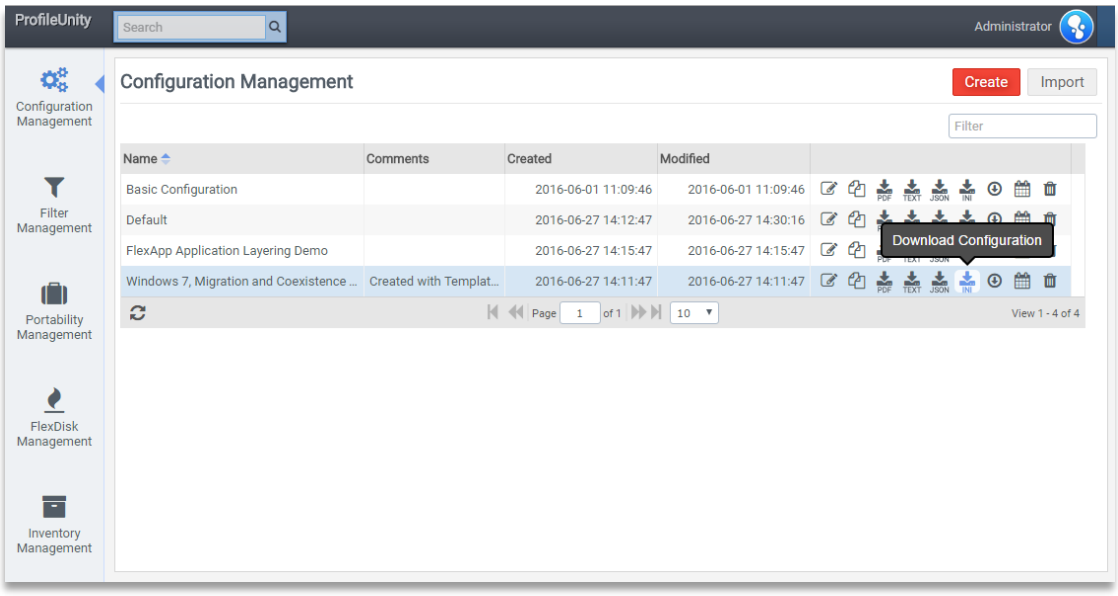
The Deployment Path can also be specified in the global settings when you login to the ProfileUnity Management Console for the first time:



The Deployment Path is again displayed as the last step in the Guided Configuration Wizard. From here, you can deploy the new Configuration as long as you are logged in as an Active Directory user. To have ProfileUnity copy the Configuration INI file to the specified **Deployment Path**, click on the **Deploy Configuration** button. Or, if you are logged in as a non-Active Directory user, admins can download and manually copy the Configuration INI to the proper location by clicking on the **Download Configuration** button.

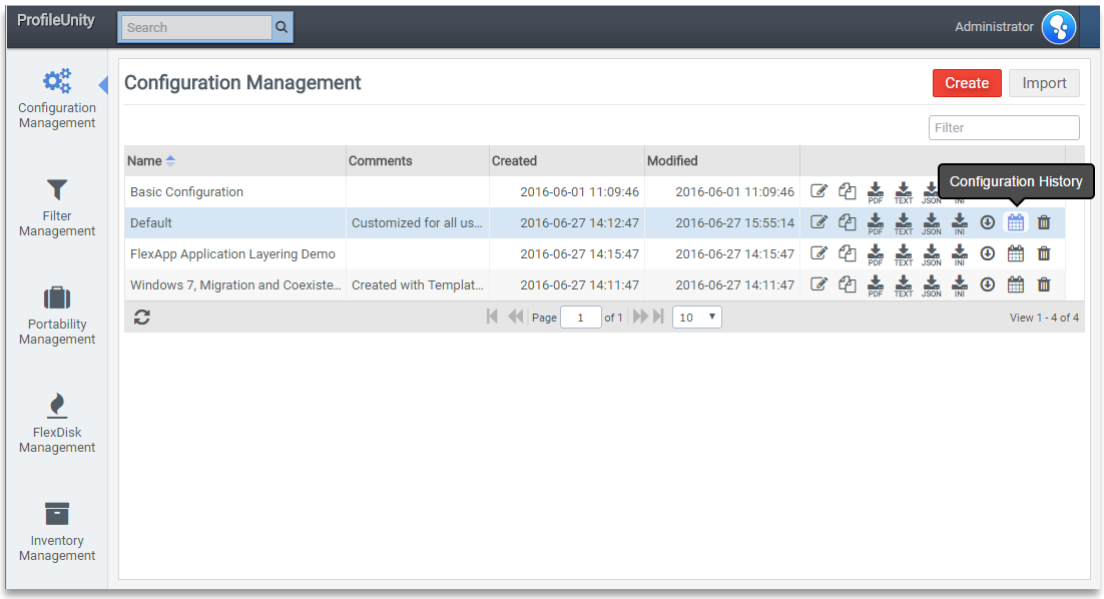


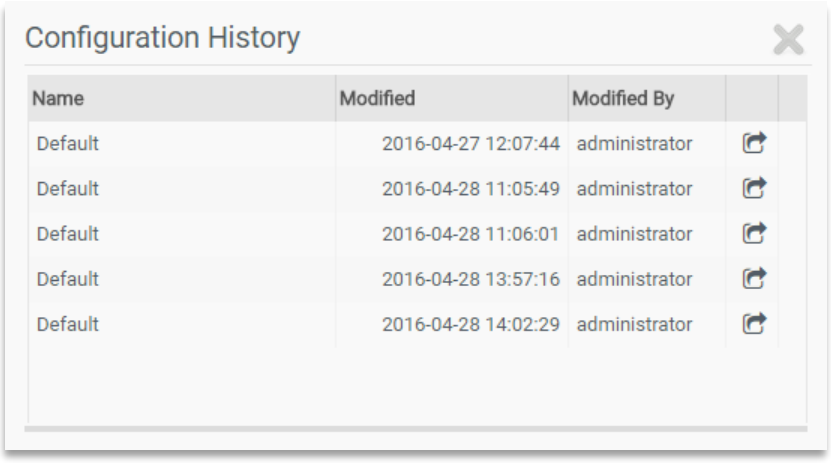
Or if you still prefer to download and copy Configuration INI files to the Deployment Path yourself, click on the **Download Configuration** icon to the right of the name of the Configuration in the Configuration Management List and then copy it to the correct location.



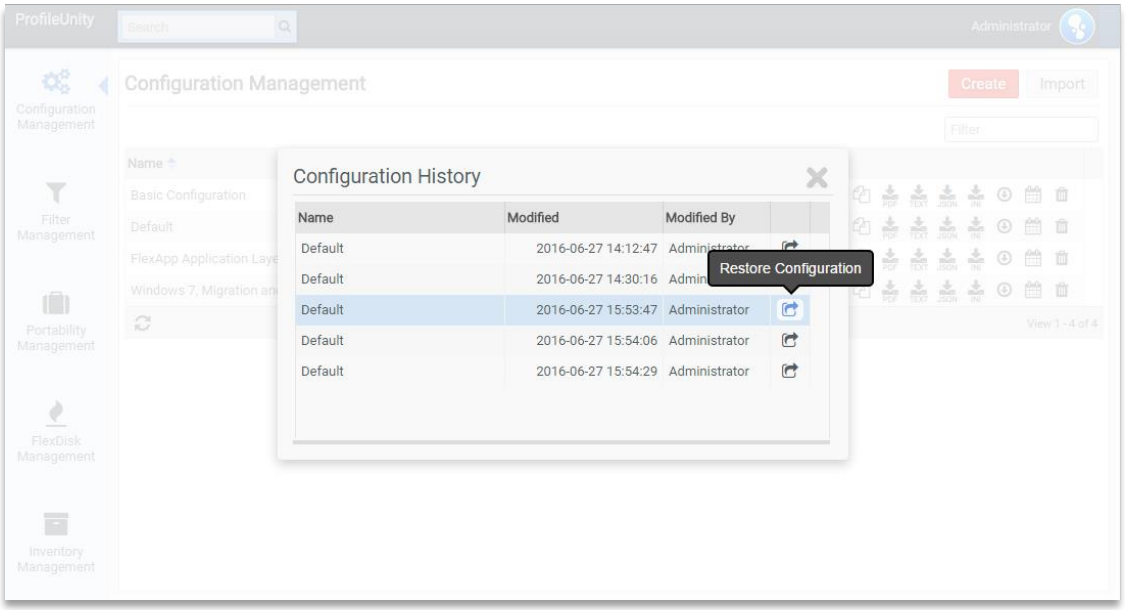
Using the Configuration Rollback Feature

As part of the configuration deployment feature, ProfileUnity has a built-in retention policy. Anytime that a Configuration file is updated, it will be saved. By default, ProfileUnity will keep track of the last five updates to each Configuration. To change the number of Configuration files to save, modify the **Configuration Revision Retention** setting in the Inventory area of the Administration Settings. You can view the list of saved changes by clicking on the **Configuration History** icon to the right of each Configuration name in the Configuration Management View.





To restore any of the last five versions of a particular Configuration, click on the **Restore Configuration** icon to the right of the Configuration date and time stamp.



Clicking on the **Restore Configuration** icon will create a new Configuration in the Configuration Management List that will start with the original Configuration name and end with "Restored" and the date and time stamp of the restored configuration. The older Configuration will be restored to the Management Console Configuration Management list. However, you will still need to deploy the restored Configuration to the Deployment Path in order for it affect your user's settings.

Anytime you delete an existing Configuration in the Management Console or need to restore an older version, remember to also go to your environment's Deployment Path to manually delete the Configuration INI files that no longer need to be applied in your environment.

Maintaining Security While Changing User Settings

ProfileUnity executes in the security context of the user logging on to the network. Privileges are never elevated during execution, ensuring your security policies are preserved. Maintaining security is not without cost. ProfileUnity is only able to modify settings the user can modify. You will need to consider the impact that security will have on your configuration. For example, attempting to modify a registry value that the user does not have permission to will fail.

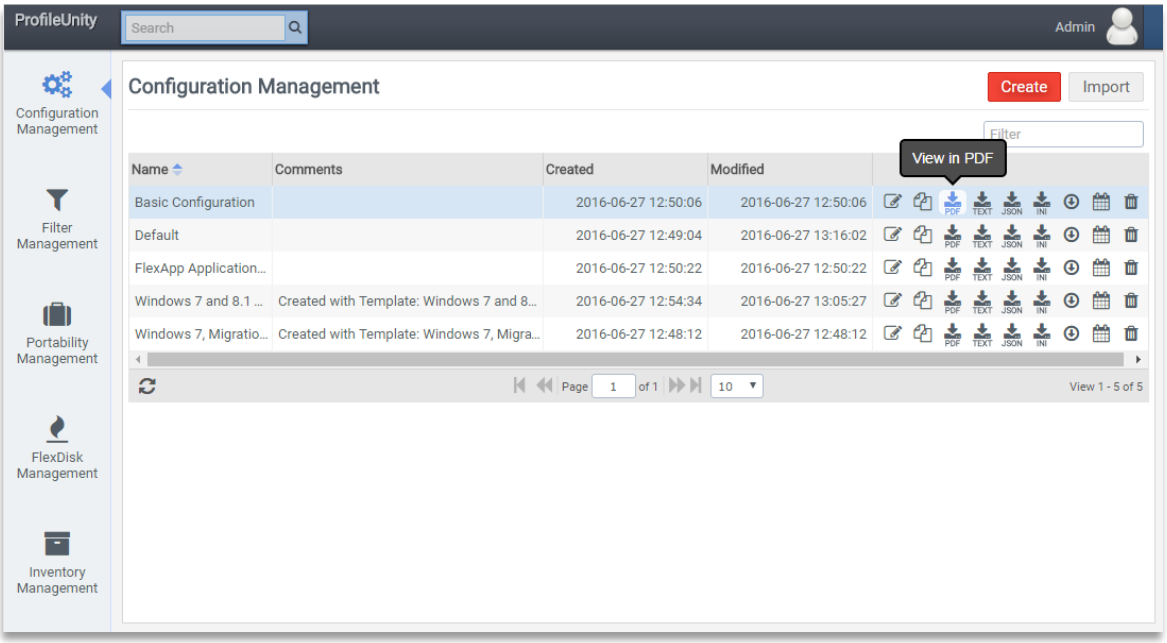
Working with Multiple Configurations

Depending on the needs in your environment, ProfileUnity can be setup to run one or more configurations. Multiple configurations allow your settings to be grouped into logical boundaries. For example, in a large organization with multiple IT departments, each department can have its own configuration. Members of one department can modify their settings without disturbing the settings of any other department. Configurations can also be layered to enforce an order of operations. For example, you can map drives in one configuration and then access objects on those drives in a subsequent configuration.

When ProfileUnity executes, it obtains the list of available configurations from the authenticating domain controller. If the configuration **Default** is present, it is always processed first. The remaining configurations are then processed in alphabetical order.

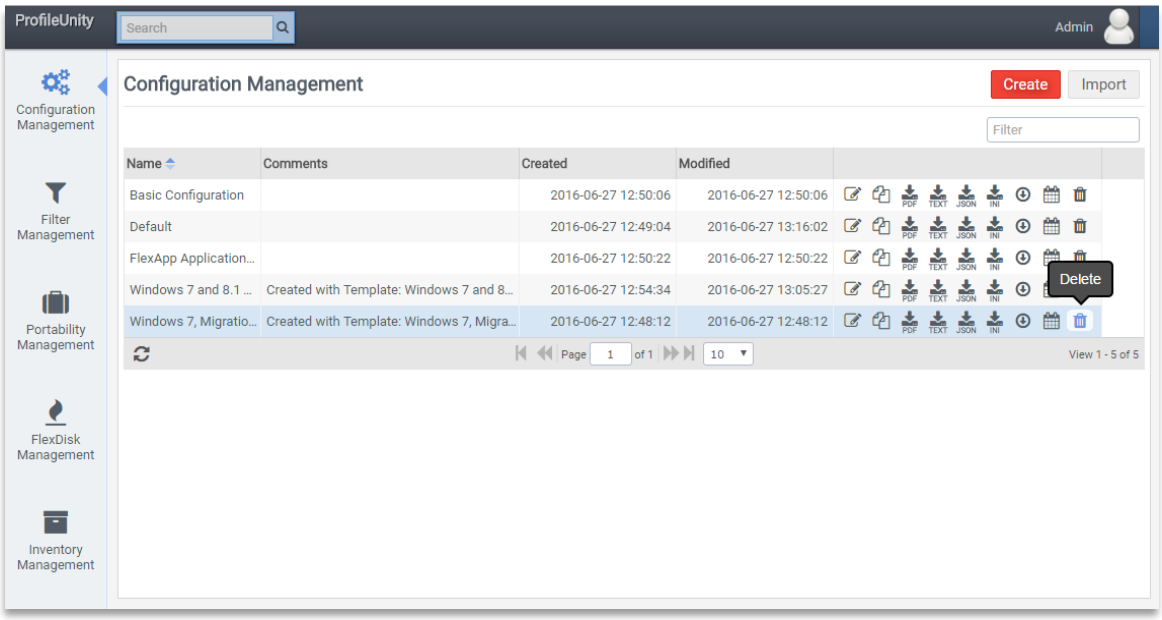
Generating a Configuration Summary Report

With the mission of decoupling user settings and data from a device's operating system, ProfileUnity offers a high level of customization. Documenting all the settings for each configuration can easily be done from the Configuration Management user interface. Simply go to the **Configuration Management** area, and click on either the **View in PDF** or **View in Text** icon next to the name of the configuration for which you need a report to download and save the report in either a Portable Document Format (PDF) or text format. These configuration reports can also be very helpful in troubleshooting issues.



Deleting Configurations

Removing old or unused configurations is easily done from the Configuration Management user interface. Simply select the **Delete** trash can next to the name of the configuration you want to remove. As a safeguard, ProfileUnity will ask you to confirm the deletion. Please note that once the configuration is deleted, it cannot be recovered.



Utilizing Filter Management

Filters in ProfileUnity are used to limit the scope of configuration rules. By using filters, you can further refine how settings are applied based on real segmentations that already exist in your environment such as different department groups or physical machine users versus virtual machine users. In addition, roaming users benefit from ProfileUnity's context aware filters which can be set up to dynamically configure their settings based on their location as they move throughout the building.

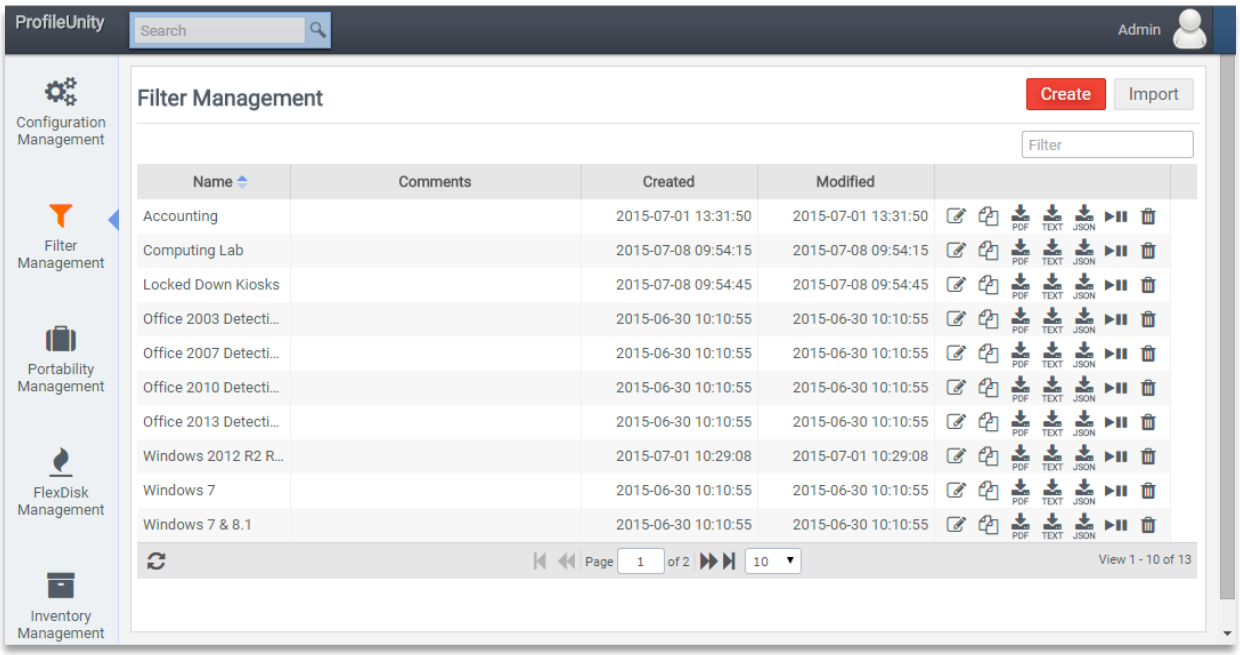
As configuration rules are processed during ProfileUnity's execution, the filter assigned to each configuration rule is evaluated. If the filter evaluation returns true, the configuration rule is applied. If the filter evaluation returns false, the configuration rule is skipped. Thus, filters allow a single ProfileUnity configuration to be applied differently to multiple client machines and users.

To give you a better idea of when you would use filters, here are some examples.

One real segment within corporations is different departments. Suppose you want to map the G: drive to the finance share on your file server for all employees in your Accounting department. You could first create a filter that tests for membership in the Accounting group and then assign this filter to the configuration rule that maps the G: drive to your file server's finance share. The filter will only allow the G: drive to be mapped for members of the Accounting group rather than everyone in the company.

In roaming environments such as hospitals, call centers and schools, you may want to change resource settings based on a user's location. For example, in a classroom environment, students move from classroom to classroom or go to computer labs in different locations. Having a long list of all possible printers to choose from would create a great deal of confusion. Instead, you could create a context aware filter that limits printer choices and even sets the default printer based on where the student is located at login.

Creating, modifying, deleting, and summarizing filters is done through the Filter Management area of the ProfileUnity Management Console.



Creating a New Filter

If you have not setup any filters, your Filter Management list will be empty. To create a new filter for ProfileUnity, click on the red **Create** button and choose your options to setup your custom filter.

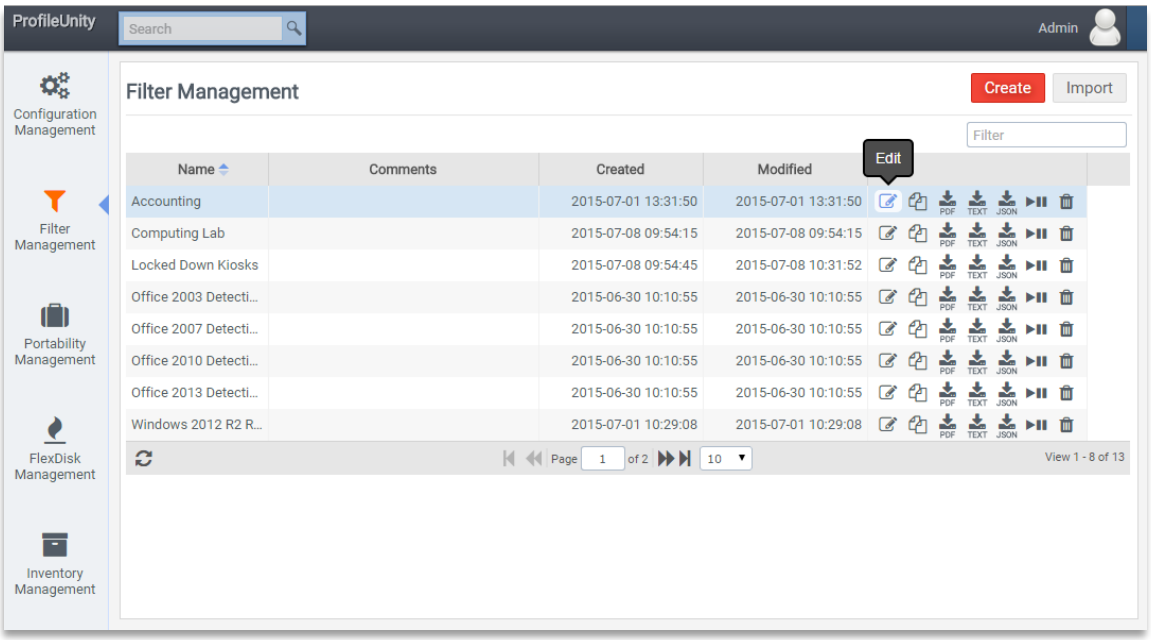
If you already have one or more filters in the Filter Management list, you may also click on **Copy**  next to a filter to clone an existing filter to use as a starting point for a new filter. Clicking **Copy** will duplicate the filter settings and put the new filter in edit mode allowing you to give the new filter a name and make any other changes to your settings.

Editing Filters

To change a filter's settings, select one the following options next to your particular filter in the Filter Management list:

 **Edit** – Modifies an existing filter

 **Copy** – Duplicates an existing filter



Filters consist of the following elements: a filter name, filter rules, machine classes, operating system platforms, connection types and system events.

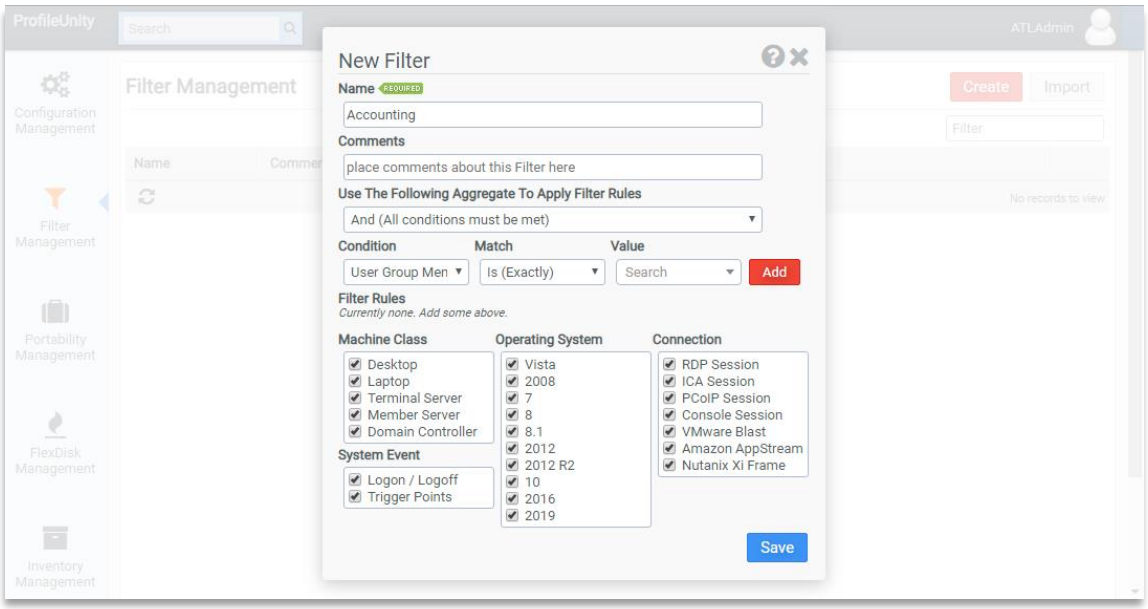
For more information on the use of specific Filter Conditions, please see "Appendix A – Filter Rule Conditions & Examples".

Adding and Removing Filter Rules

Filter rules are comprised of a **Condition**, **Match**, and **Value** triplet. Choose the **Condition** the filter is testing for from the drop-down list. ProfileUnity offers filtering based on conditions like user name, computer name, IP address and many more. Similarly select the appropriate **Match** option from the drop-down list. The **Match** is a Boolean operator that specifies the relationship you are testing between the **Condition** and the **Value**. Once you enter the **Value** to test for, click the **Add** button to create the new filter rule and display it in the Filter Rules list.

A filter can have multiple filter rules. The logic used to evaluate multiple filter rules is based on the **Use The Following Aggregate To Apply Filter Rules** (**Rule Aggregate**) option. If the **Rule Aggregate** option is set to AND (all conditions must be met), then all rules must evaluate true for the filter to return true. If the **Rule Aggregate** option is set to OR (any one condition must be met), the filter will return true if any rule evaluates true. The **Rule Aggregate** option applies to all rules in a filter. It is not possible to specify AND logic for some rules and OR logic for other rules.

To delete a filter rule, select **Remove** to the right of the particular rule triplet in the Filter Rules list.



Selecting Machine Classes

Filters can include or exclude different machine classes. Available choices are:

- **Desktop**
- **Laptop**
- **Terminal Server**
- **Member Server**
- **Domain Controller**

The Desktop and Laptop machine classes will return true for all client machines that are not Terminal Servers, Member Servers, or Domain Controllers.

Selecting Operating Systems

Filters may include or exclude different operating systems. ProfileUnity works with:

- **Windows Vista**
- **Windows 7**
- **Windows 8**
- **Windows 8.1**
- **Windows Server 2008**
- **Windows Server 2012**
- **Windows Server 2012 R2**
- **Windows 10**
- **Windows Server 2016**
- **Windows Server 2019**

Selecting Connection Types

Filters can include or exclude different network connection types such as:

- **RDP Session**
- **ICA Session**
- **PCoIP Session**
- **Console Session**
- **VMware Blast**
- **Amazon AppStream**
- **Nutanix Xi Frame**

Selecting System Events

Filters can be applied at different times based on certain events. Choose from:

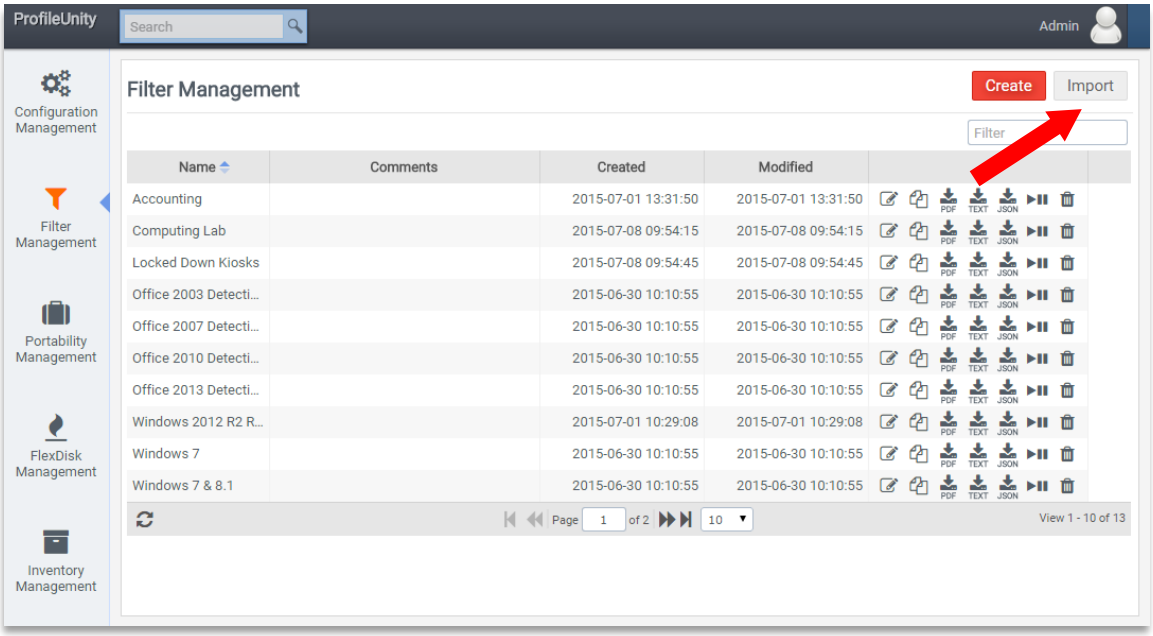
- **Logon/Logoff**
- **Trigger Points**

Saving or Discarding Your Changes

When you have finished editing your filter, click on the **Save** button to save all your changes before leaving the Filter Editor. Click on the 'X' in the top right corner to cancel your changes. All unsaved changes will be discarded.

Importing and Exporting Filters

Individual filters can also be imported into the ProfileUnity Management Console. Just click on the **Import** button at the top right of the Filter Management area and specify the location of the JavaScript Object Notation (JSON) filter file. The new filter will be added to the Filter Management list. Please note that filters must have unique names. The Management Console will not import filters that have the same exact name of a filter that is already in the Filter Management list.



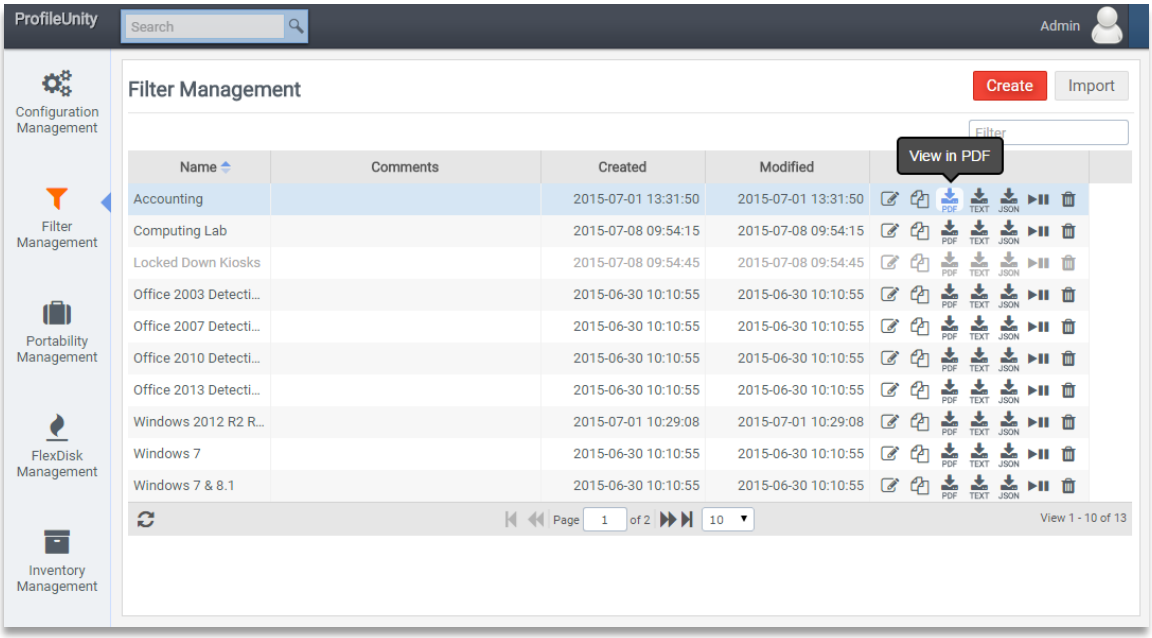
To export or backup an existing filter just click on the **Export Filter** JSON button next to the filter name and specify the location where the JSON file should be saved.

Disabling or Enabling Filters

Filters can be disabled or enabled for use by the ProfileUnity configuration files by toggling the **Disable/Enable** button in the Filter Management list. All filters are enabled by default. The filter name will be grayed out when it is disabled.

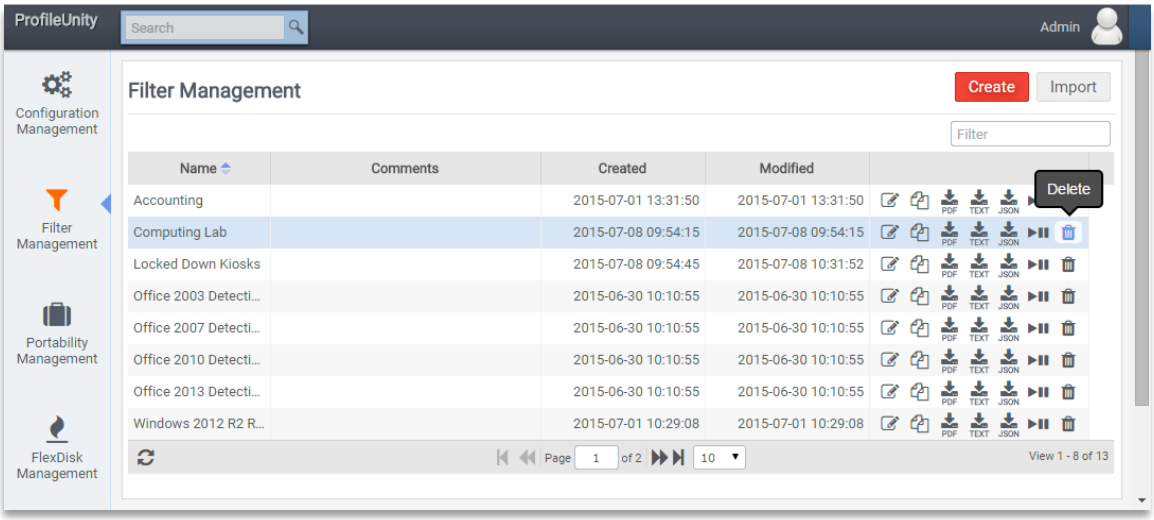
Generating a Filter Summary Report

Documenting all the filter settings available for each configuration can easily be done from the Filter Management user interface. Simply go to the **Filter Management** list, and click on the **View in PDF** or **View in Text** icon next to the name of the filter for which you need a report to download and save the report in either a Portable Document Format (PDF) or text format. These filter reports can also be very helpful in troubleshooting issues.

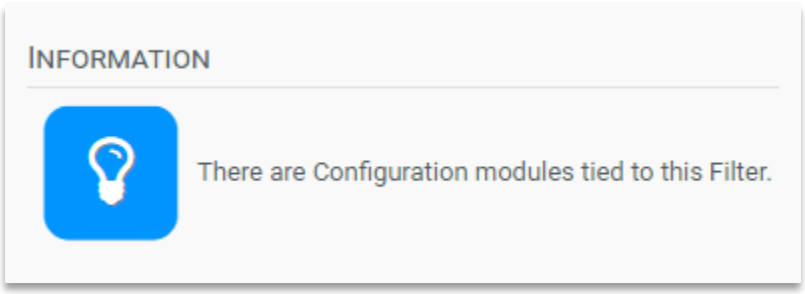


Deleting Filters

Removing old or unused filters is easily done from the Filter Management list. Simply select **Delete** next to the name of the filter you want to remove. As a safeguard, ProfileUnity will ask you to confirm the deletion. Please note that once the filter is deleted, it cannot be recovered.



If you attempt to delete a filter still in use by one or more of your configurations, you will receive the warning shown below and the filter will not be deleted.



Anytime you make a change to a configuration, including filter settings, you must re-download or re-deploy the INI file. In addition, those changes will not take effect until the next logon event from the user. If they are currently logged in when the configuration file is downloaded, they will still be under the old configuration settings.

Providing Profile Portability & Management

What happens to the user profile when you have a mixed environment of physical and virtual machines running different operating systems and possibly Remote Desktop Services sessions? ProfileUnity bridges the gap by truly decoupling user profiles and data from the operating system allowing seamless movement between different versions of Windows operating systems while using one, unified profile per user rather than one profile per machine type or operating system. ProfileUnity's heterogeneous environment support allows users to log in from anywhere on any Windows device and have their user settings and data follow them wherever they go.

ProfileUnity solves the difficulties in making personal user profile data available across multiple Windows sessions. When a user logs on to a Windows session, his/her personal user profile settings are instantly pulled across the network in seconds. Customized settings such as application level customization, user-created spell checker data, Outlook signatures, desktop wallpaper, and much more are instantly made available for the user regardless if the machine is a VMware View, Citrix XenDesktop, thin client, or traditional Windows desktop.

But ProfileUnity takes profile portability one step further by allowing you to manage what parts of the profile are made portable. By default, a large amount of data is stored in a user's profile. However, not all of this data is necessary for a robust and complete user experience. ProfileUnity is granular by design, allowing you to set rules and choose what profile data is to be made portable. This makes for speedy logon times, reduces profile corruption instances, and eliminates the needless transfer of large amounts of data over the network.

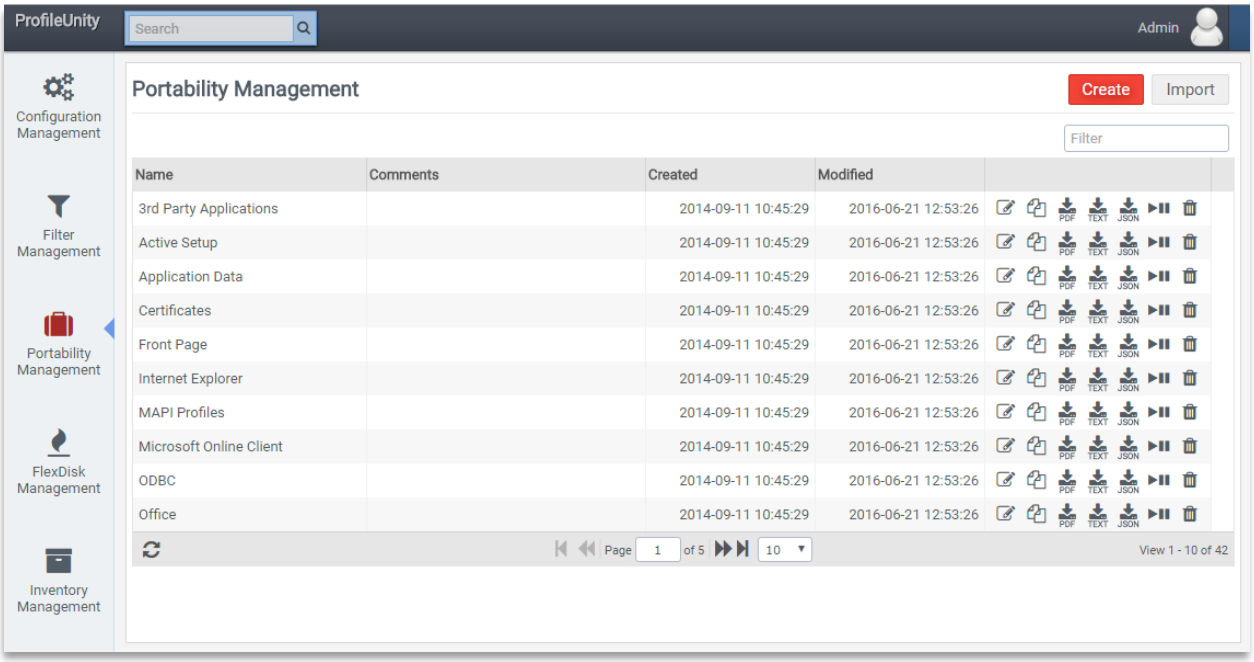
Can't decide which parts of the user profile to make portable? Are user login times growing along with profile sizes? Are you experiencing performance issues syncing large user files, such as Outlook data files, or a large number of files kept in a file repository? Starting with version 6.5, ProfileUnity offers a second option for profile portability called ProfileDisk. ProfileDisk allows a user's entire local Windows profile to be stored on a VHD or VMDK and attached to the desktop as a separate layer when the user logs in. You won't have to spend time deciding which parts of the user's profile should be saved and made portable across your environment in an effort to reduce profile sizes. And since the user profile is available after mounting the drive at login rather than waiting for it to be streamed across the network, large profiles load much faster for your users.

But combining ProfileDisk with Portability Management really gives you the best of both worlds. User profiles will load faster and be easier to manage while administrators will still have granular control over moving data even if another application uses non-standard practices in creating user output. We'll cover more about ProfileDisk later and cover the basics of Portability Management here.

With ProfileUnity, making user profile data portable is a two-step process. First, you define rulesets through the Portability Management interface. Rulesets control the portions of the user profile that will follow the user around the network. Second, once your rulesets are defined, the **Portability Settings** configuration module is used to control how user profile data is stored and retrieved from the network. Please see **Appendix C – Configuration Modules** for assistance with the **Portability Settings** configuration module.

ProfileUnity includes predefined rulesets that are designed to manage popular settings. These predefined rulesets can be used to quickly setup portability with minimal configuration. You can modify these predefined rulesets as well as create your own rulesets.

Creating, modifying, deleting, and summarizing rulesets is done through the Portability Management area of the ProfileUnity Management Console.



Creating a New Portability Ruleset

To add a custom profile portability ruleset in addition to the predefined rulesets included with ProfileUnity, click on the **Create** button in the Portability Management list. You may also click on **Copy** next to an existing ruleset to clone an existing ruleset to use as a starting point for a new ruleset. Clicking **Copy** will open the Ruleset Editor with a copy of your existing ruleset allowing you to give the new ruleset a name and make any other changes to your settings.

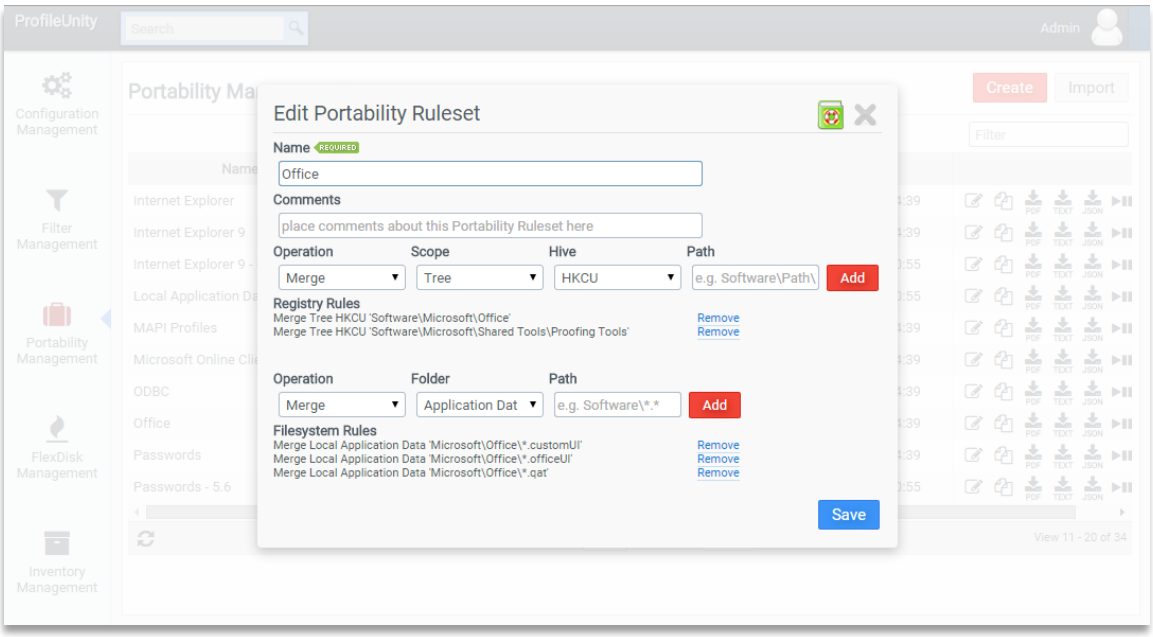
Editing Rulesets

The Ruleset Editor is where all profile portability ruleset configuration takes place. To edit a ruleset, select one the following options in the Portability Management list:

Edit – Modifies an existing portability ruleset

Copy – Duplicates an existing portability ruleset

User profiles are comprised of two types of data: registry settings and files and folders. Rulesets handle these data types with registry and file system rules respectively. Rulesets consist of the following elements: a ruleset name, registry rules, and file system rules. As you will see from the predefined rulesets, some rulesets contain only registry rules, some contain only file system rules, and others contain both types of rules.



Adding and Removing Registry Rules

Registry rules are comprised of an **Operation**, **Scope**, **Hive**, and **Path**. Choose the **Operation** the ruleset performs from a drop-down list. When saving and restoring registry settings, ProfileUnity can Merge, Replace or Exclude data. The following lists the **Operation** options to choose from:

Registry Rule Operation	Description
Merge	Saved data is merged with existing data during restore.
Replace	Existing data is replaced with saved data during restore.
Exclude	Data is excluded from save and restore.

Similarly select the appropriate **Scope** of the operation from a drop-down list. The **Scope** can be limited to a registry Tree, Key, or Value. These are the **Scope** options to choose from:

Registry Rule Scope	Description
Tree	Path specifies a key. Specified key and values, subkeys, and subkey values.
Key	Path specifies a key. Specified key and values only, no subkeys or subkey values.
Value	Path specifies a value. Specified value only.

Next select the appropriate **Hive** that the registry rule applies to from a drop-down list. These are the **Hive** options to choose from:

Registry Rule Hive	Description
HKCU	HKEY_CURRENT_USER registry root key
HKLM	HKEY_LOCAL_MACHINE registry root key

Once you enter the **Path** of the registry item, click the **Add** button to create the new registry rule and add it to the Registry Rules list.

> ***The Path is relative to the Hive or registry root key you select. Your entries should not begin with "HKEY_CURRENT_USER" or "HKEY_LOCAL_MACHINE" since ProfileUnity adds this portion of the path by design.***

A ruleset can have multiple registry rules. ProfileUnity registry rules allow granular storage and retrieval of registry values.

To delete a registry rule, select **Remove** to the right of the particular rule.

Registry Rule Example

Operation	Scope	Hive	Path
Merge	Tree	HKCU	Software\Microsoft\Office
This registry rule will save and restore the registry values contained in the `HKEY_CURRENT_USER\Software\Microsoft\Office` key and all subkeys. During restore, existing data will be overwritten with the saved values.			

Adding and Removing File System Rules

File system rules are comprised of an **Operation**, **Folder**, and **Path** triplet. Choose the **Operation** the ruleset performs from a drop-down list. When saving and restoring file system settings, ProfileUnity can Merge, Replace or Exclude data. The following lists the **Operation** options to choose from:

File System Rule Operation	Description
Merge	Saved data is merged with existing data during restore.
Replace	Existing data is replaced with saved data during restore.
Exclude	Data is excluded from save and restore.

Similarly select the appropriate shell **Folder** from a drop-down list. Here are your options to choose from:

File System Rule Folder	Description
Application Data	Specified path is relative to the Application Data shell folder.
Cookies	Specified path is relative to the Cookies shell folder.
Desktop	Specified path is relative to the Desktop shell folder.
Favorites	Specified path is relative to the Favorites shell folder.
History	Specified path is relative to the History shell folder.
My Documents	Specified path is relative to the My Documents shell folder.
Program Group	Specified path is relative to the Programs shell folder.
Recent	Specified path is relative to the Recent shell folder.
Send To	Specified path is relative to the SendTo shell folder.
Start Menu	Specified path is relative to the Start Menu shell folder.
Startup Group	Specified path is relative to the Startup shell folder.
User Profile	Specified path is relative to the user profile root folder.
Local Application Data	Specified path is relative to the Local AppData shell folder.
Program Files	Specified path is relative to the %programfiles% environment variable.
System Drive	Specified path is relative to the %systemdrive% environment variable.
System Root	Specified path is relative to the %systemroot% environment variable.
My Music	Specified path is relative to the My Music shell folder.
My Pictures	Specified path is relative to the My Pictures shell folder.
My Video	Specified path is relative to the My Video shell folder.
Temporary Internet Files	Specified path is relative to the Temporary Internet Files shell folder.
NetHood	Specified path is relative to the NetHood shell folder.
PrintHood	Specified path is relative to the PrintHood shell folder.
Templates	Specified path is relative to the Templates shell folder.

File System Rule Folder	Description
Links	Specified path is relative to the Links shell folder.
Contacts	Specified path is relative to the Contacts shell folder.
Searches	Specified path is relative to the Searches shell folder.
Roaming Tiles	Specified path is relative to the Roaming Tiles shell folder
Libraries	Specified path is relative to the Libraries shell folder.
Custom	Specified path is relative to a custom shell folder.

Enter the **Path** which specifies the folder that contains the user profile data. The **Path** is relative to the **Folder** that was selected. Click the **Add** button to create the new file system rule.

> ***File system rules with an empty Path component include all files/folders under the entry specified in Folder.***

A ruleset can have multiple file system rules. File system rules allow granular storage and retrieval of files and folders located in the user profile.

To delete a file system rule, select **Remove** to the right of the particular rule triplet.

File System Rule Example

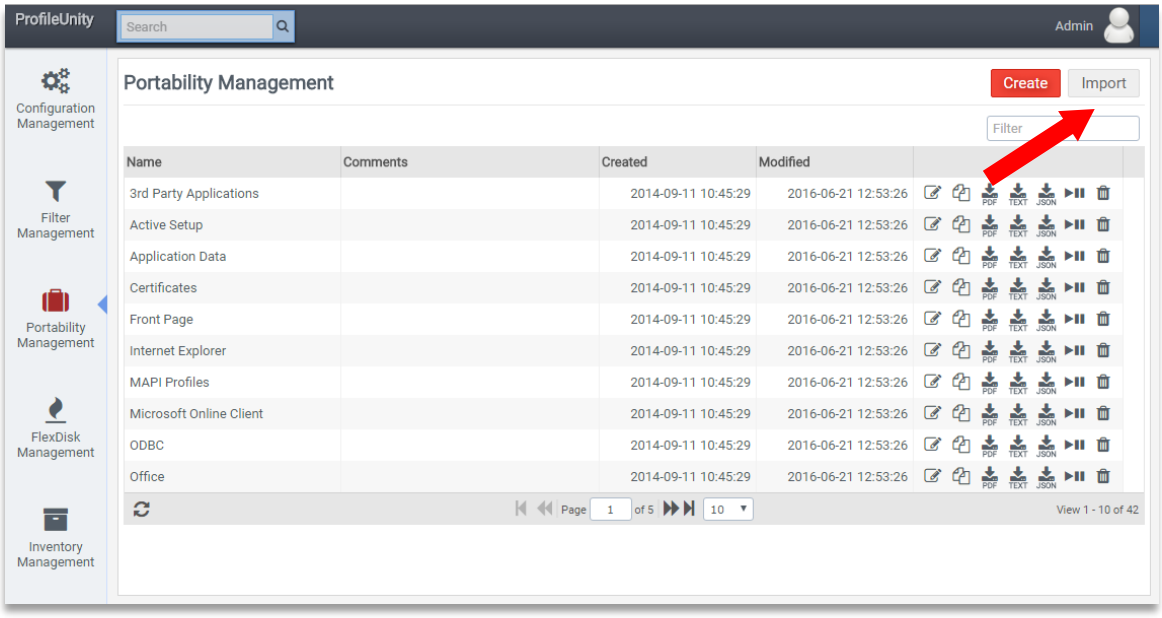
Operation	Folder	Path
Merge	Application Data	Microsoft
This file system will save and restore the files/folders contained in the `Microsoft` folder located in the `Application Data` shell folder. During restore, existing data will be overwritten with the saved data.		

Saving or Discarding Your Changes

When you have finished editing your ruleset, click on the **Save** button to save all your changes before leaving the Ruleset Editor. Click on the 'X' in the top right corner to cancel your changes. All unsaved changes will be discarded.

Importing and Exporting Portability Rulesets

Individual rulesets can also be imported into the ProfileUnity Management Console. Just click on the **Import** button at the top right of the Portability Management area and specify the location of the JavaScript Object Notation (JSON) portability settings file. The new ruleset will be added to the Portability Management list. Please note that portability rules must have unique names. The Management Console will not import rulesets that have the same exact name of a ruleset that is already in the Portability Management list.



To export or backup an existing ruleset:

1. Go to the **Portability Management** area and locate the name of the ruleset in the list.
2. Select the **Export Portability Ruleset** option from the ruleset's row. 

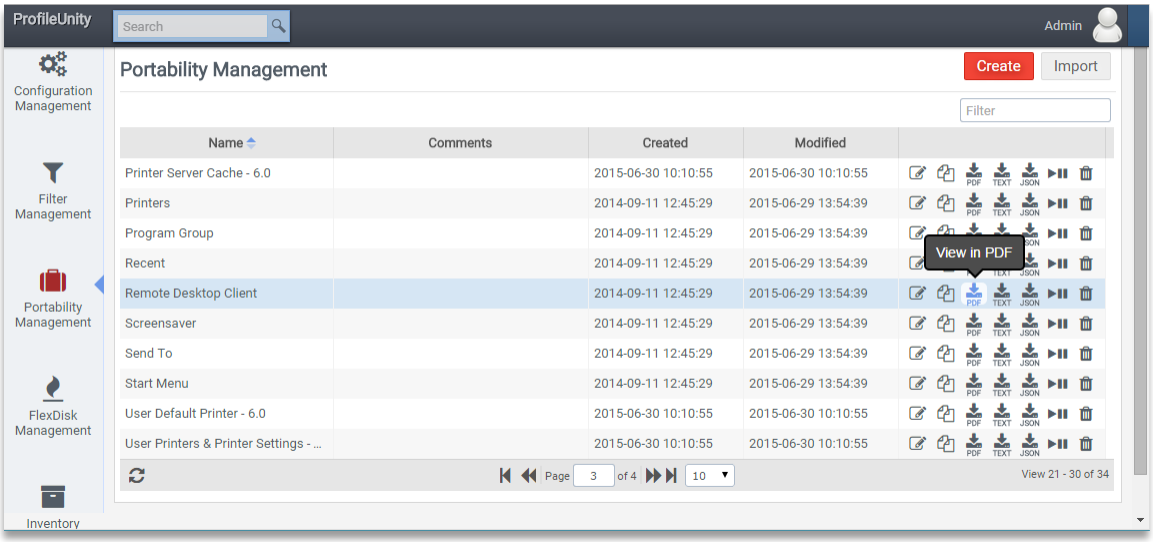
3. Specify the location where the JSON file should be saved.

Disabling or Enabling Portability Settings

Rulesets can be disabled or enabled for use by the ProfileUnity configuration files by toggling the **Disable/Enable** icon in the Portability Management list. All rulesets are enabled by default. The ruleset name will be grayed out when it is disabled.

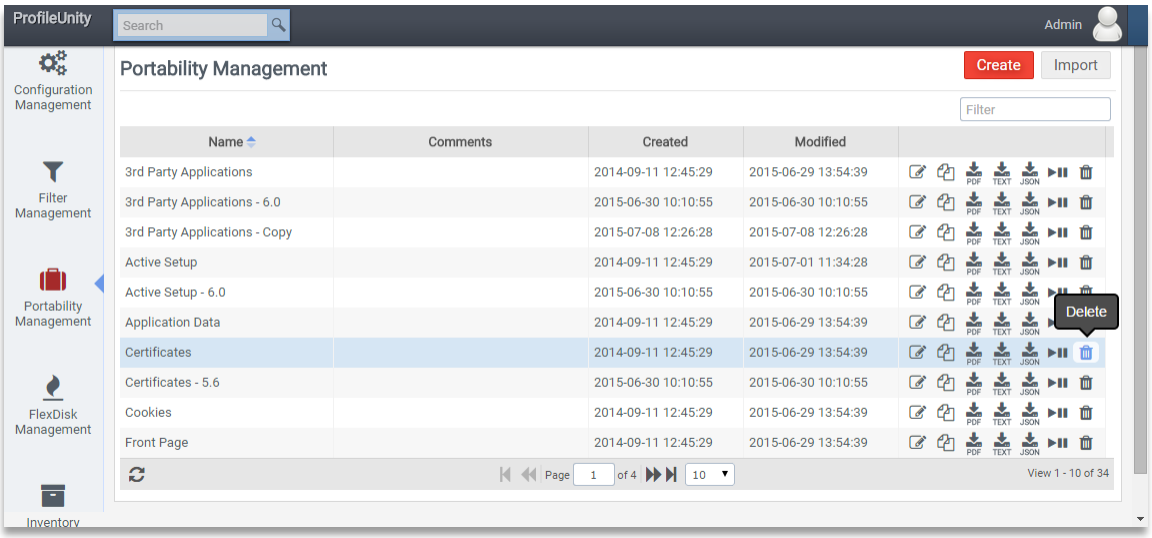
Generating a Portability Summary Report

Documenting all the portability settings available for each configuration can easily be done from the Portability Management user interface. Simply go to the **Portability Management** list, and click on **View in PDF or View in Text** next to the name of the ruleset for which you need a report to download and save the report in either a Portable Document Format (PDF) or text format. These ruleset reports can also be very helpful in troubleshooting issues.

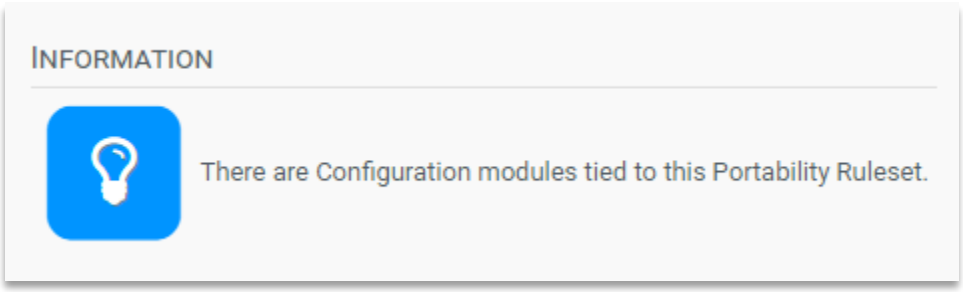


Deleting Portability Rulesets

Removing old or unused rulesets is easily done from the Portability Management user interface. Simply select **Delete** next to the name of the ruleset you want to remove. As a safeguard, ProfileUnity will ask you to confirm the deletion. Please note that once the ruleset is deleted, it cannot be recovered.



If you attempt to delete a portability ruleset still in use by one or more of your configurations, you will receive the warning shown below and the ruleset will not be deleted.



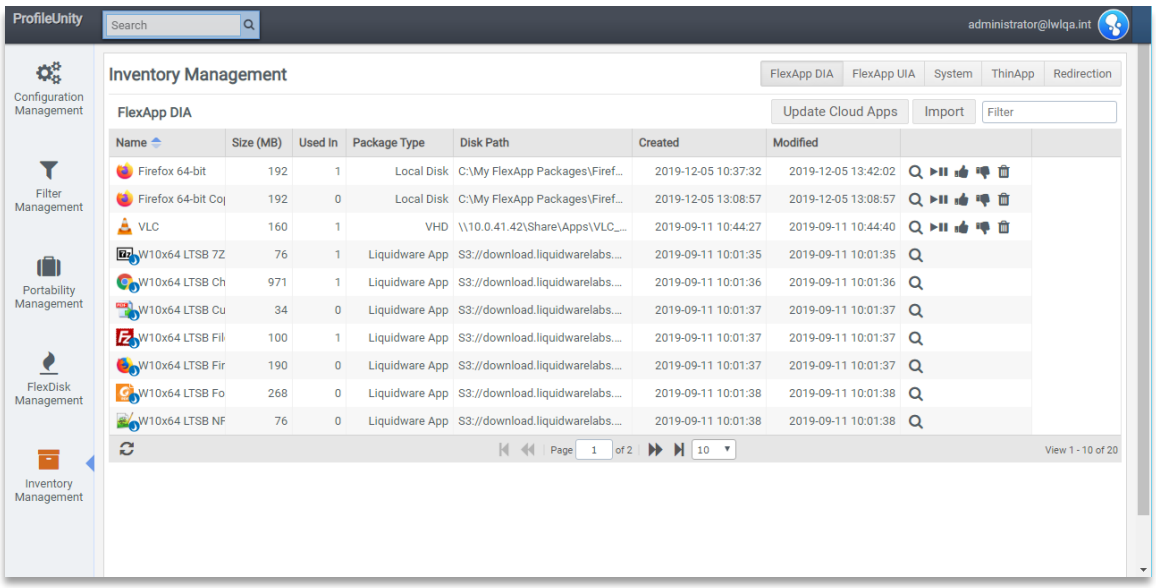
Anytime you make a change to a configuration, including portability rule settings, you must re-download or re-deploy the INI file. In addition, those changes will not take effect until the next logon event from the user. If they are currently logged in when the configuration file is downloaded, they will still be under the old configuration settings.

Inventory Management

ProfileUnity's Inventory Management was designed to provide administrators with an overall view of FlexApp Packages, System Inventory, ThinApp Packages, and Redirection in the environment.

FlexApp DIA

The FlexApp DIA inventory shows admins all of the FlexApp DIA packages available for use in the FlexApp library. From this list, admins may view specific package information, delete packages, or disable/enable packages. Note that all functions will not be available for Liquidware Cloud Apps, since those are owned and maintained by Liquidware and made available to all FlexApp users.

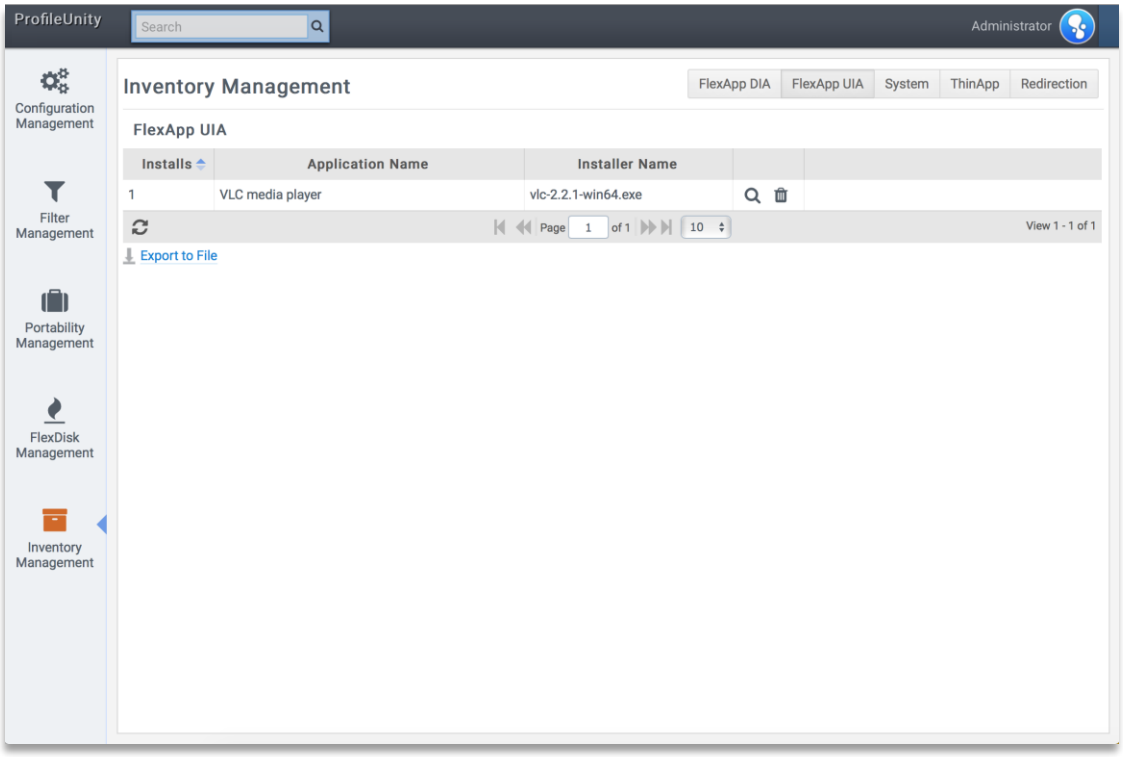


Use the **Update Cloud Apps** button at the top to pull the most recent versions of Liquidware Cloud Apps that are available on cloud storage for public use.

Use the **Import** button at the top to browse servers and import additional FlexApp DIA packages.

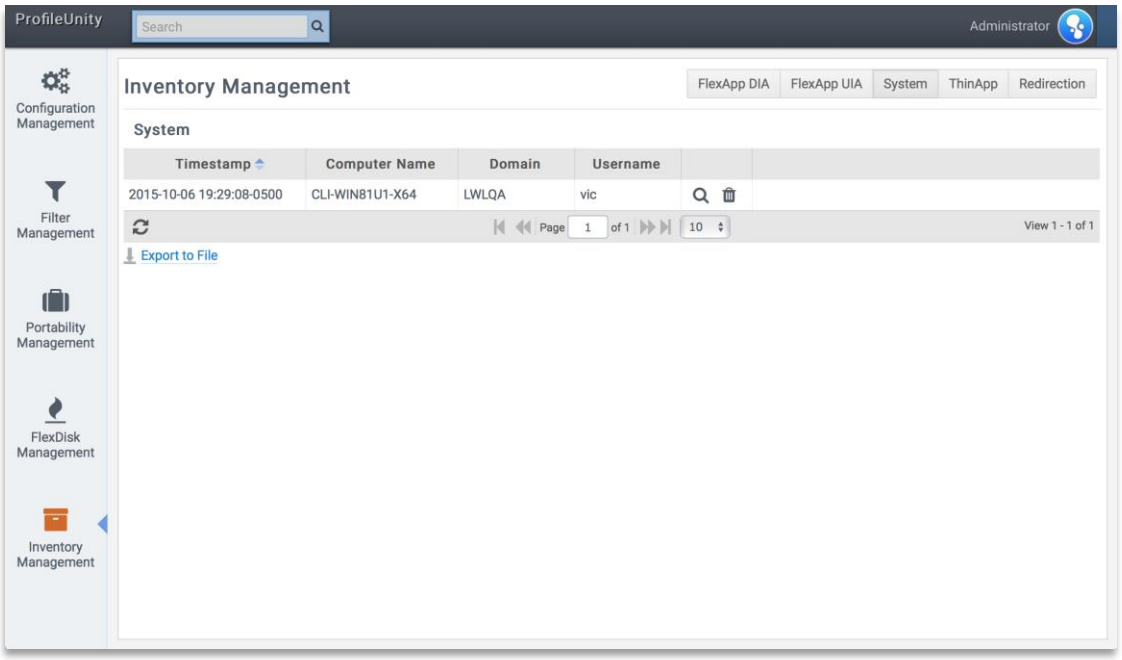
FlexApp UIA

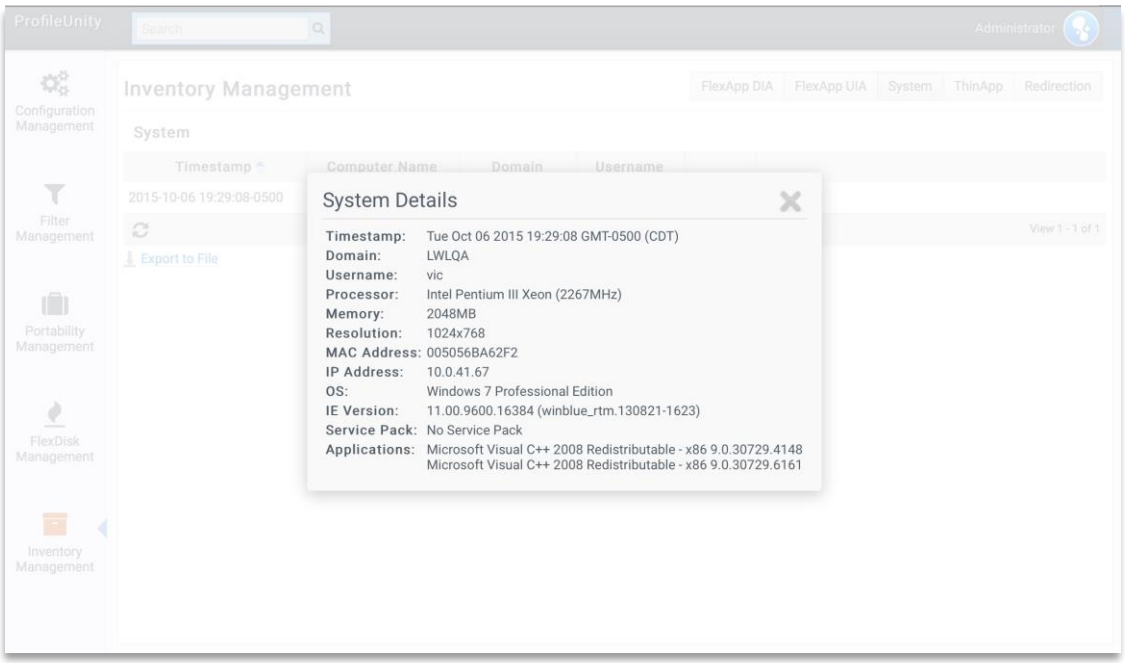
The FlexApp UIA inventory provides an administrator with the ability to report on what their users are installing with FlexApp UIA. This inventory will tell you what application was installed and who installed.



System Inventory

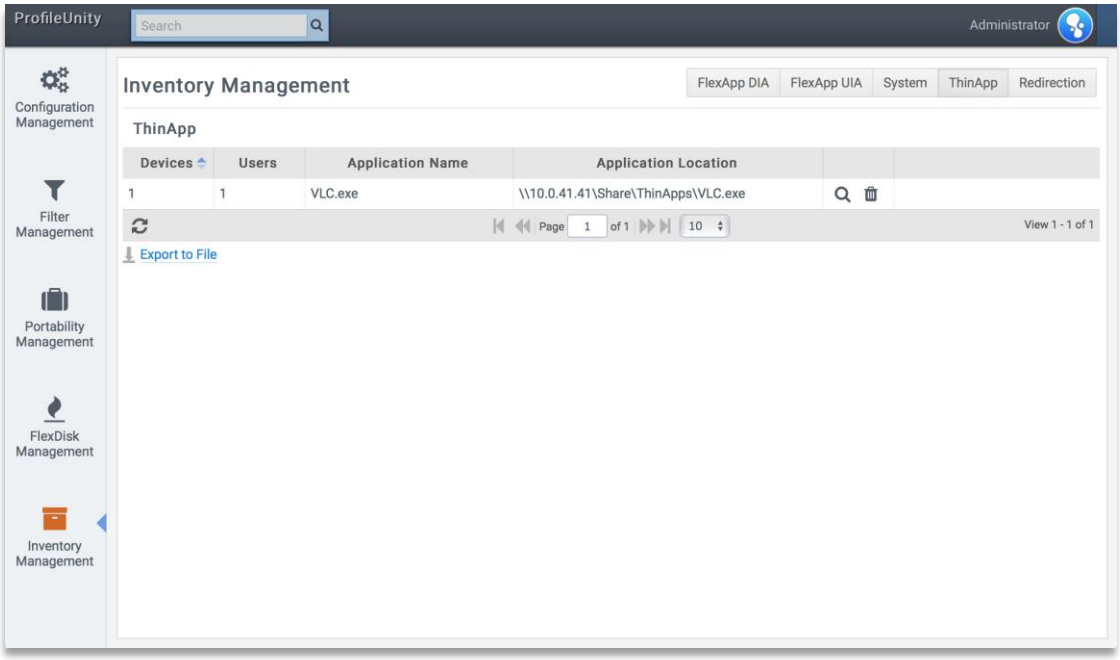
The System inventory displays an in-depth hardware and software inventory report for the chosen record.





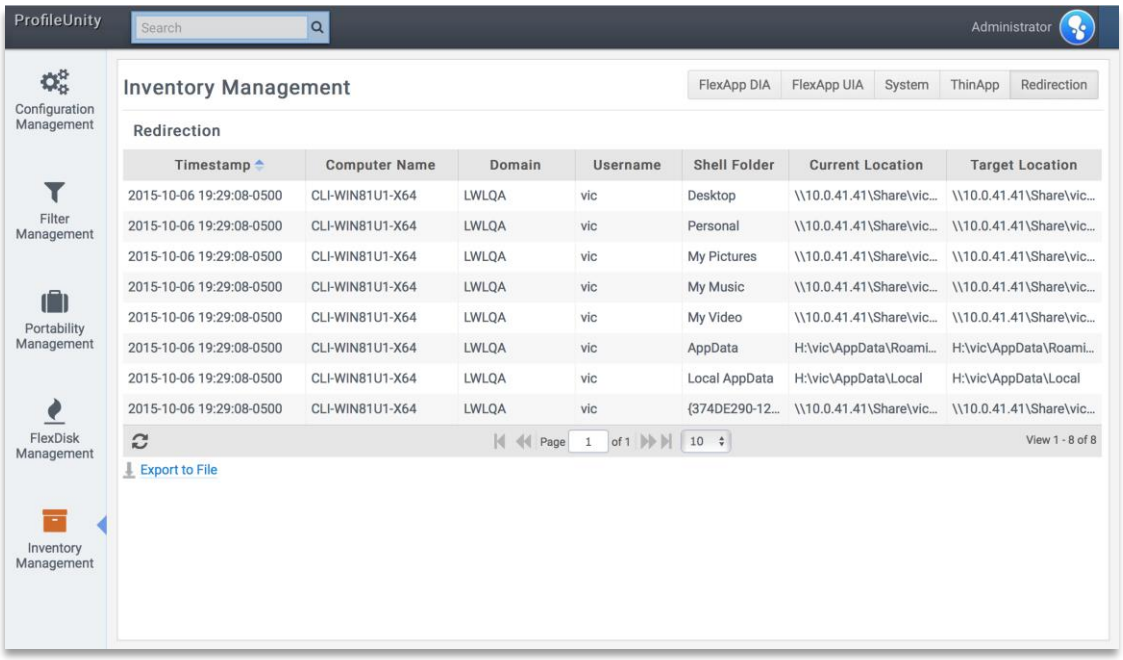
ThinApp Inventory

The ThinApp inventory displays an in-depth report for each ThinApp package including the ThinApp app name, how many computers, users and path to the ThinApp package.



Redirection

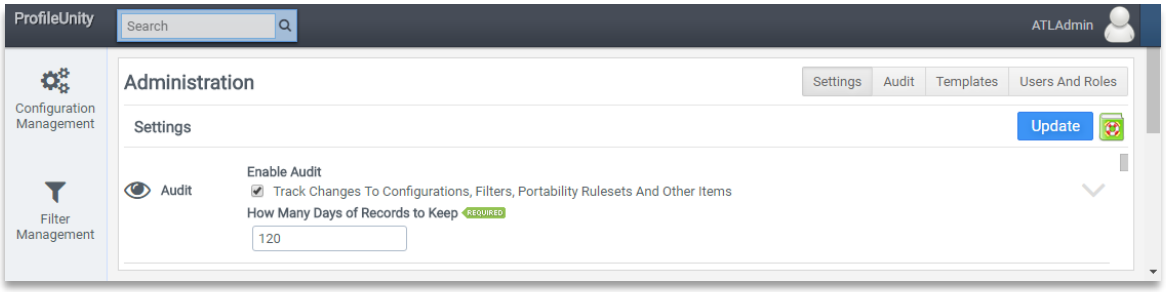
The Folder redirection inventory will tell you the status of a user's shell folder during migration. For example, a user's "My Documents" folder could be in the process of synchronizing, finished with synchronizing or stalled due to an error. This way an administrator knows when a user's shell folders have finishing synchronizing to the target path.



Changing Administration Settings

The Administration Settings area allows you to control various settings and operations for ProfileUnity in your environment. To get to these settings, go to your login user ID at the top right of the Management Console interface and select **Administration**. At the top of the Administration area, select **Settings**. As you scroll through the list you will see options under the following categories: Audit, Cloud Storage, Global Variables, Client Settings, ProfileUnity Tools, ProfileUnity Console Service Log, Proxy, Miscellaneous, WebServices, Database, Inventory, Clustering, FlexDisk, License Reporting, and Notifications. Please note that the Authentication category has moved to the **Administration > Users and Roles** tab.

Audit

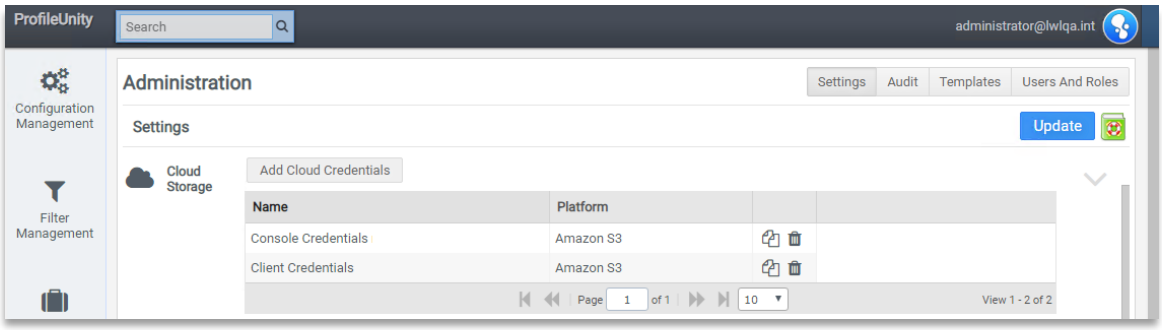


Enable Audit:
Check **Track Changes to Configurations, Filters, Portability Rulesets And Other Items** in order for ProfileUnity to keep an audit trail of revisions to your database.

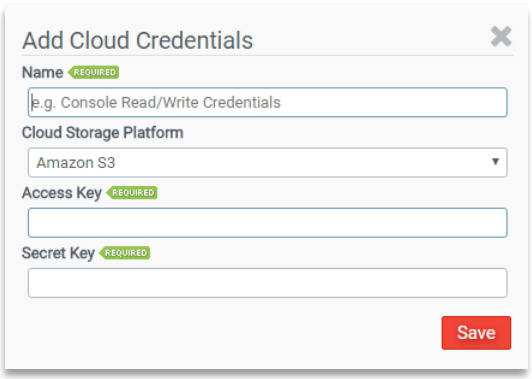
How Many Days of Records to Keep:
If enabling audit tracking, enter the number of days to retain audit information. The default is 120 days.

Cloud Storage

ProfileUnity supports the use of cloud storage for configuration INI files and user profiles.



To make use of cloud storage spaces, click on the **Add Cloud Credentials** button.



Name:
Enter your descriptive name for this cloud storage connection. You will need at least two sets of credentials per cloud storage platform – one for the ProfileUnity Console functions and one for the ProfileUnity Client functions.

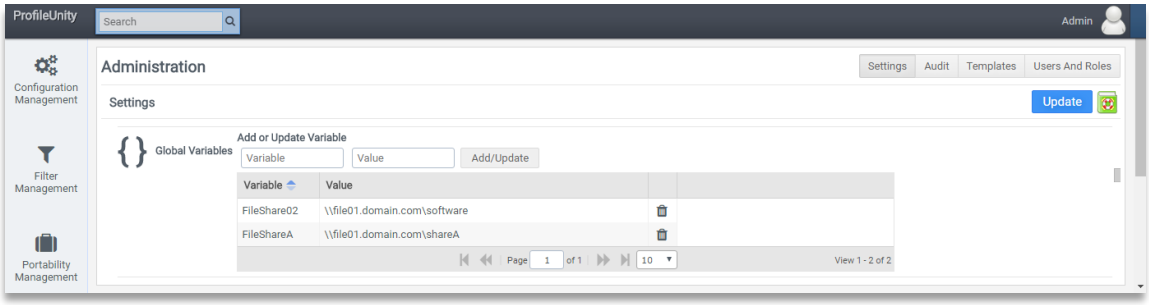
Cloud Storage Platform:
Select the cloud storage provider from **Amazon S3**, **Microsoft Azure**, or **Google Cloud Platform**.

Access Key:
Enter your user access key for this cloud storage connection.

Secret Key:
Enter the secret key for this cloud storage connection.

Global Variables

Administrators have the ability to use global variables in across all objects in the Management Console. This is a time-saving feature that allows for global changes to occur just by changing the value of variable once for all occurrences. For example, one organization might set a global variable for a storage path called “FileShareA”. If the path that this variable points to changes in the future, just edit the global variable and the new value will be substitute across Configurations, Filters, and Portability Rulesets.



To setup global variables for use in your environment:

Enter a **Variable Name** and a **Value**. Then click on **Add/Update Variable** to either add a new variable or update an existing value.

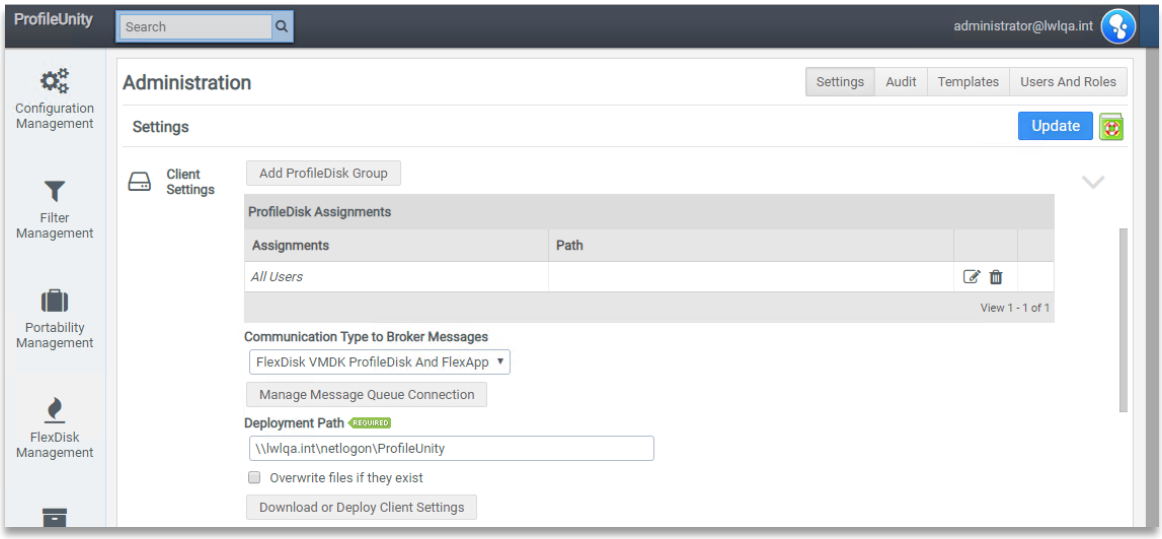
Please note that all variable names must be made up of letters and numbers only.

To use a global variable in a Configuration, Filter, or Portability Ruleset, type in the variable name surrounded by curly brackets. For example, to use a global variable named “FileShare02”, type in **`{FileShare02}`**.

> ***Anytime you make a change to a configuration, including global variables that may affect multiple configurations, you must re-download or re-deploy the related Configuration INI files. In addition, those changes will not take effect until the next logon event from the user. If they are currently logged in when the configuration file is downloaded, they will still be under the old configuration settings.***

Client Settings

This section handles client-side settings for requesting licensing, ProfileDisk VHD configurations, and FlexDisk configurations.



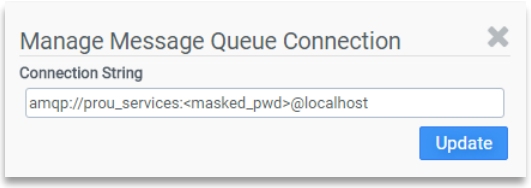
ProfileDisk allows a user's entire Windows user profile to exist on a virtual disk that can be attached and detached to a desktop as needed. The ProfileDisk stores 100% of all file system and registry changes in the user profile path so that all user settings and data are preserved on both persistent and non-persistent desktops. VHD-based ProfileDisks are managed in this section. VMDK-based ProfileDisks are managed by FlexDisk.

Add ProfileDisk Group:
Click this button to assign one or more VHD-based ProfileDisks to Active Directory User Groups. Choose the **Storage Type, Virtual Disk Path, Size in GB,** and whether the disk will be **Expandable** or **Fixed** in size. Check the **Enable Compression** option to allow space savings. Different VHD-based ProfileDisks can be assigned to particular Active Directory User Groups. Also, each group of users can have separate paths to their own ProfileDisk so that loads can be balanced among different file shares.

ProfileDisk Assignments:
Lists the VHD-based ProfileDisks in your environment by user assignment and disk location. You can edit or remove ProfileDisks from this list.

Communication Type to Broker Messages:
Select which ProfileUnity technologies will be sending messages across the Fabric. Choose **Licensing Only Fabric Use** if you will not be using ProfileDisk or FlexDisk. In this case the Fabric will just be used for license requests. Choose **VHD ProfileDisk** for licensing and VHD ProfileDisk communications. Choose **FlexDisk VMDK ProfileDisk And FlexApp** to allow both ProfileDisk and FlexApp to make use of VMDKs and to send licensing communications across the Fabric. Choose **FlexDisk VMDK ProfileDisk Only** to allow licensing and VMDK ProfileDisk communications only.

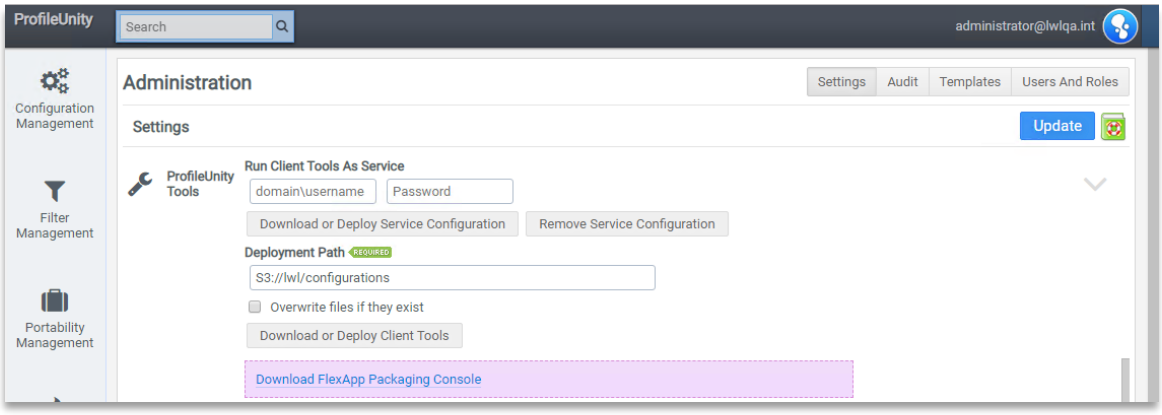
Click on the **Manage Message Queue Connection** button to edit the message broker connection string.



Connection String:
The connection string to connect the Message Broker. This field is blank by default. Do not modify this field without guidance from Liquidware Support.

Deployment Path:
Enter the deployment location where the client configuration will be stored on your NETLOGON share. Check the **Overwrite files if they exist** option to have the deployed configuration file overwrite any previously existing client configuration files at the specified location. Next, click on the blue **Update** button to save your settings. Then click on the **Download or Deploy Client Settings** button. Then you will see another dialog box with an option to pick the **Platform** for deployment. If you choose the "Domain" or "Cloud" option, this will place the client configuration file, nodes.xml, to the directory you specified for the **Deployment Path**. If you choose the "Download" option, the client configuration file will be downloaded through your browser.

ProfileUnity Tools



In addition to the ProfileUnity Management Console, there are two other main components to ProfileUnity: the FlexApp Packaging Console and the Client. The FlexApp Packaging Console allows administrators to configure and prepare any applications that will need to be configured for users and made available as a department installed application (DIA). The Client manages each user's settings and persona during their session.

Run Client Tools As Service:
Enter a username and password in the form of domain\username to run the Client tools as a service. Then click on the blue **Update** button.

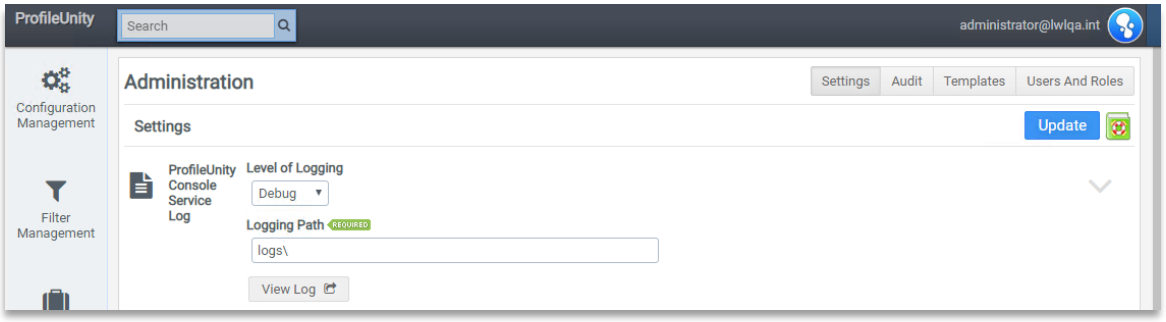
Click **Download or Deploy Service Configuration** to deploy the configuration to specified **Deployment Path**. In order for this to work, you also need the system INI path set in the GPO and the license path set to the main deployment path. Then you will see another dialog box with an option to pick the **Platform** for deployment. If you choose the "Domain" or "Cloud" option, this will place the service configuration file to the directory you specify for the **Deployment Path**. If you choose the "Download" option, the service configuration file will be downloaded through your browser.

To delete a configuration, click on the **Remove Service Configuration** button.

Deployment Path:
Enter the UNC or cloud storage path to where the client tools and configuration files should be placed. Check the **Overwrite files if they exist** option to overwrite files if they already exist at the specified location. Otherwise, an error will be thrown should the files already exist. Next, click on the blue **Update** button to save your settings. Then click **Download or Deploy Client Tools** to deploy the client tools to the deployment path. Then you will see another dialog box with an option to pick the **Platform** for deployment. If you choose the "Domain" or "Cloud" option, this will copy the Client Tools files to the directory you specified for the **Deployment Path**. If you choose the "Download" option, the Client Tools files will be downloaded through your browser.

Download FlexApp Packaging Console:
Downloads the FlexApp Packaging Console Installer to a location you specify.

ProfileUnity Console Service Log

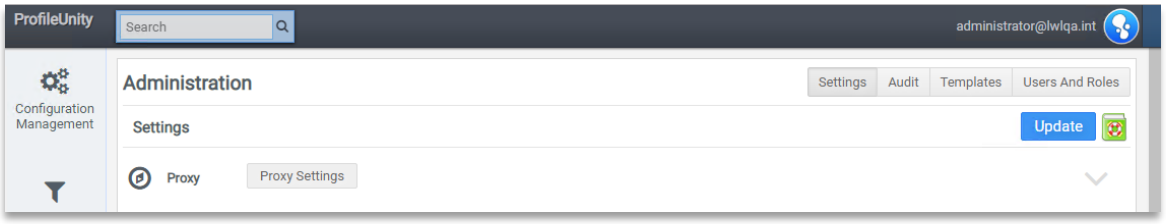


Level of Logging:
The level of logging ProfileUnity should log at. Choose from: **Debug, Info, Warning, Error,** or **Fatal**. **Debug** provides the most detailed information.

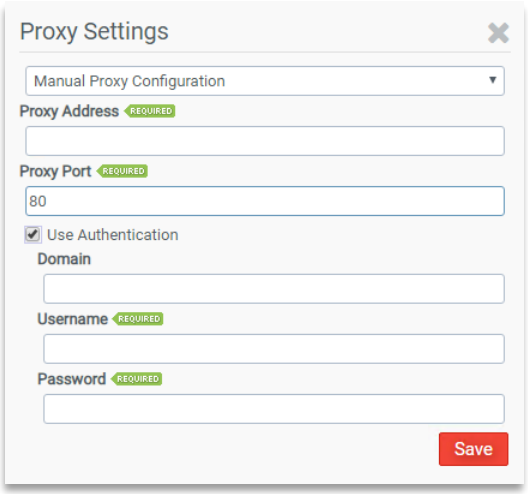
Logging Path:
The path to where ProfileUnity should log.

View Log:
View the latest log being reported.

Proxy



(Optional) Enable proxy support for cloud platforms and other calls via the internet by clicking on **Proxy Settings**.



Choose from the following options:

- Do not use Proxy
- Auto-detect Proxy
- Manual Proxy Configuration

If choosing the "Manual Proxy Configuration" option, you may use the rest of the fields.

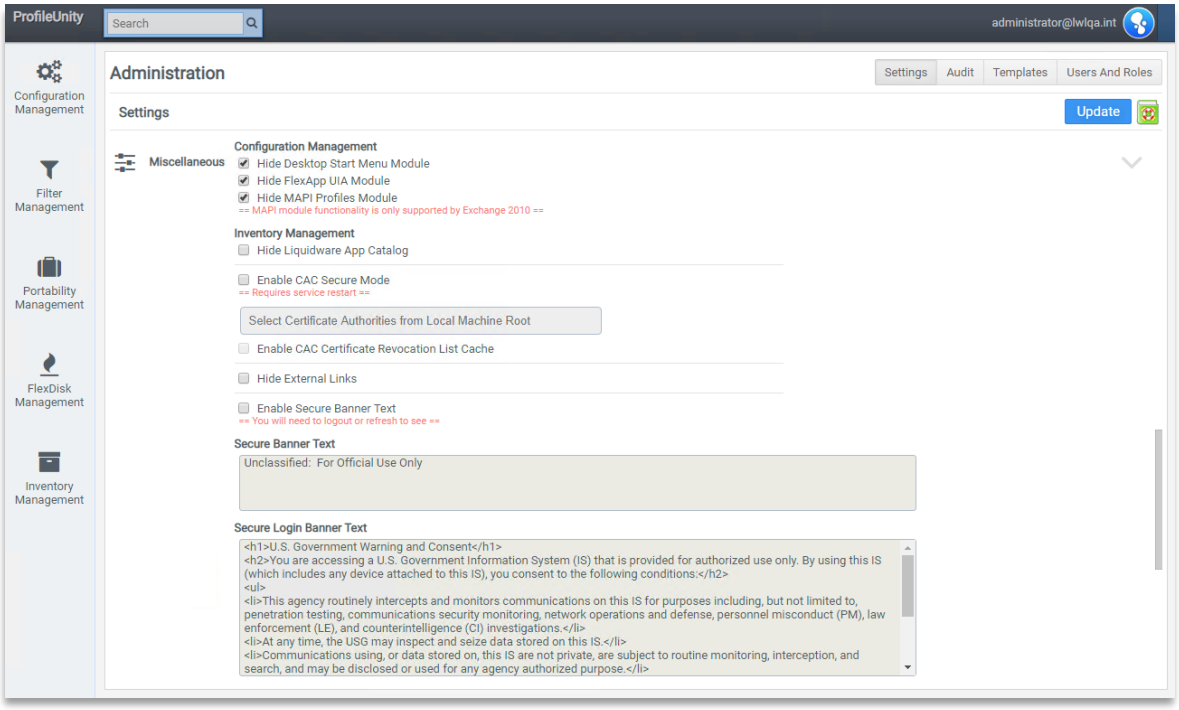
Proxy Address:
Enter the IP Address of the proxy server.

Proxy Port:
Enter the port to use for communication with the proxy server. The default is port 80.

Use Authentication:
Check this option to provide proxy authentication credentials if they are required and then complete the **Domain**, **Username**, and **Password** information.

- **Domain:** (Optional) Enter the domain for the user account information entered below.
- **Username:** Enter the username for the account to be used for authentication with the proxy server.
- **Password:** Enter the password for the account to be used for authentication with the proxy server.

Miscellaneous



Hide Desktop Start Menu Module:
Uncheck this option to allow configuration of the Desktop Start Menu module. When this option is unchecked, the Desktop Start Menu configuration module will be made available for use in the Configuration Management area.

Hide FlexApp UIA Module:
Uncheck this option to allow configuration of FlexApp User Installed Applications (UIA). When this option is unchecked, the FlexApp UIA configuration module will be made available for use in the Configuration Management area. If this option is checked, the FlexApp UIA module will remain hidden. Please note the following FlexApp UIA limitations. FlexApp UIA is not supported on Windows 10, Windows Server 2016, or on persistent desktops with any OS. FlexApp UIA and FlexApp DIA should not be used together on a user's desktop. They can be used separately.

Hide MAPI Profiles Module:
Uncheck this option to allow configuration of the MAPI Profiles module. When this option is unchecked, the MAPI Profiles configuration module will be made available for use in the Configuration Management area. Please note that the MAPI Profiles module functionality is only supported by Microsoft Exchange 2010.

Hide Liquidware App Catalog:
Check this option to hide Liquidware App cloud FlexApps from the FlexApp DIA Inventory inside both the Management Console and the FlexApp Packaging Console. Liquidware Apps are FlexApp packages that have been created by Liquidware and are immediately available for use in your environment. They are provided via cloud storage. 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 these FlexApps be cloned before using in your own environment.

Enable CAC Secure Mode:
Check this option to require Common Access Card (CAC) usage for authentication when logging in to the ProfileUnity Management Console. Get more details in our ***Configuring Common Access Card Authentication*** document.

Enable CAC Certificate Revocation List Cache:
Check this option to enable Common Access Card (CAC) certificate revocation list cache.

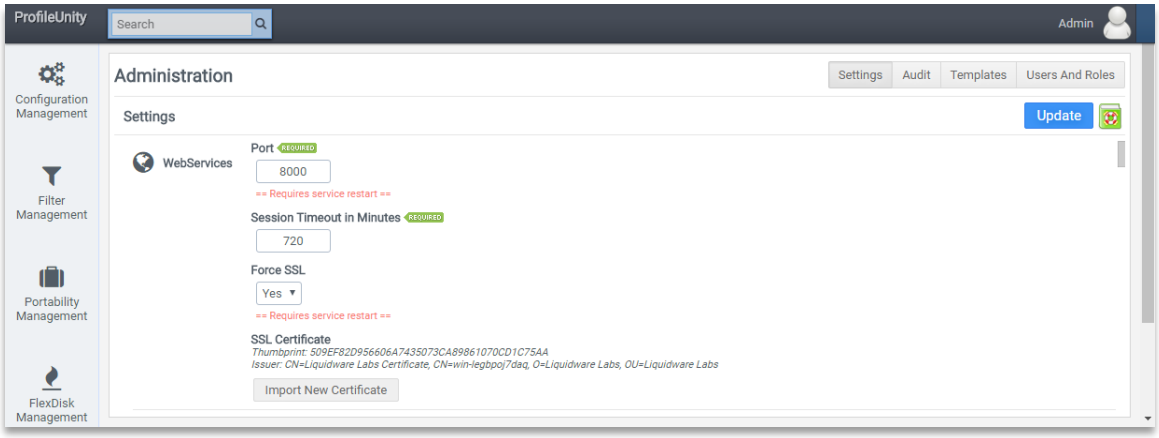
Hide External Links:
Hides external web links from the management console.

Enable Secure Banner Text:
Check this option to enable the display of all secure banner text.

Secure Banner Text:
Type the approved secure text that is displayed before a secure mode login.

Secure Login Banner Text:
Type the approved secure login text to be shown prior to, or as part of, the ProfileUnity Management Console login process.

WebServices



Port:
The port on which the web service runs. The default is 8000. Any change to the port number requires a service restart.

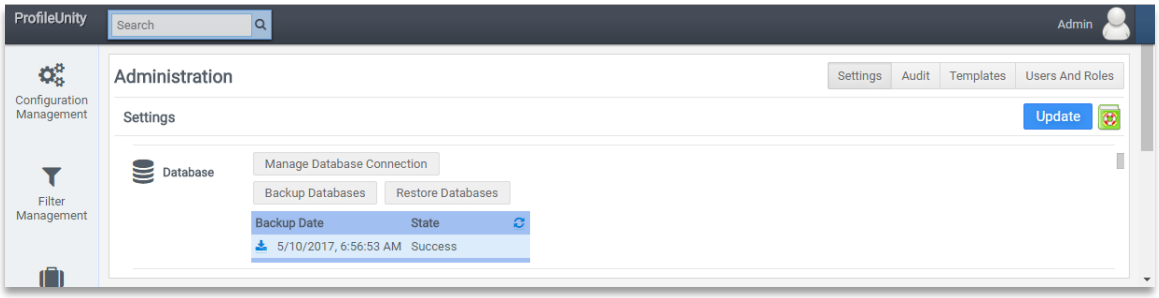
Session Timeout in Minutes:
How long before an idle user has before they will be required to log in again.

Force SSL:
Forces use of SSL HTTPS protocol over standard HTTP protocol. Any change to this setting requires a service restart.

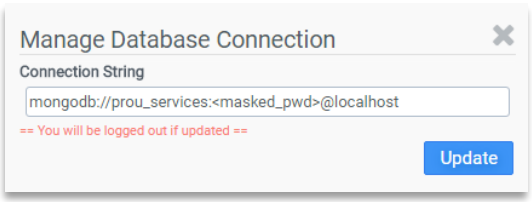
SSL Certificate:
Displays current SSL certificate.

Import New Certificate:
Allows the user to import a .PFX certificate.

Database



Click on the **Manage Database Connection** button to configure the location of your ProfileUnity database.



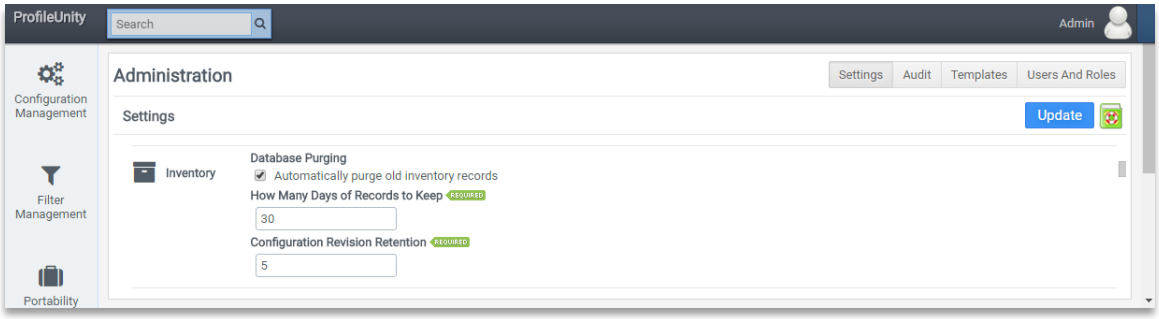
Connection String:
The connection string used to connect to the database.

> ***Note: You may use the Connection String to switch between databases by just changing the file name if you have multiple databases that you would like to maintain separately.***

You are encouraged to backup your configuration on a regular basis. To create a copy of your database, click on the **Backup Current Database** button. A list of backups will be kept below the button. Use the **Download** icon to download a specific database backup.

You can also restore your configuration settings from a previously made backup. This is especially useful if you are upgrading your software version and would like to keep your existing configuration settings. To restore an archived database, click on the **Restore Database** button and browse to the location of the backed-up database file.

Inventory



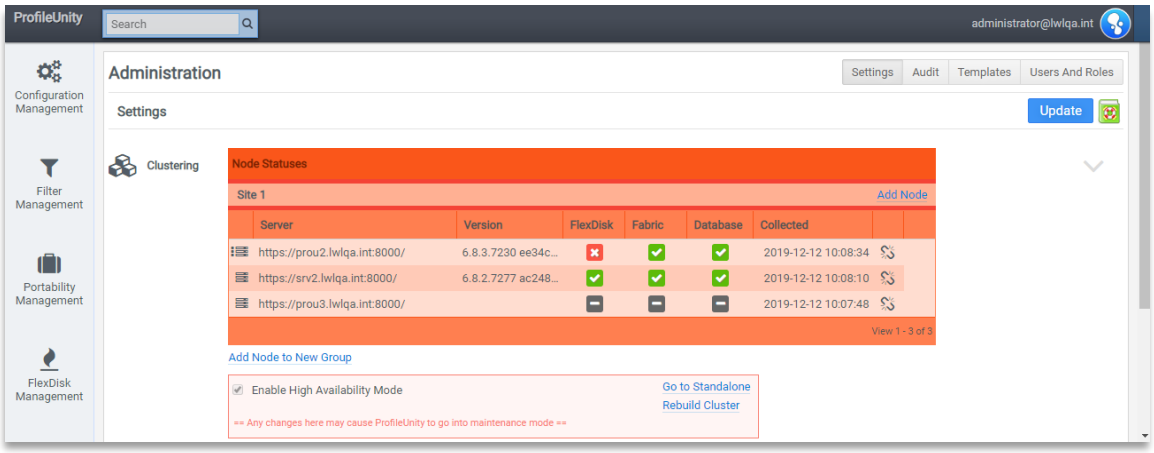
Database Purging:
Check **Automatically purge old inventory records** if you want ProfileUnity to periodically check for old records that can be deleted.

Enter how long you wish to keep inventory records in days in the **How Many Days of Records to Keep** field. ProfileUnity will automatically purge records outside of this date range if automatic purges have been enabled.

Configuration Revision Retention:
As part of the configuration auto-deployment feature, ProfileUnity has a built-in retention policy. Anytime that a Configuration file is updated, it will be saved. Set the retention number to indicate the last number of updates to each Configuration that ProfileUnity will keep. The default is 5. You can view the list of saved changes by clicking on the **Configuration History** icon to the right of each Configuration name in the Configuration Management View.

Clustering

Clustering is part of the FlexDisk technology solution where ProfileUnity is configured in a clustered mode to provide multiple nodes for scaling additional resources and to protect against a single point of failure. Clustering settings go hand in hand with FlexDisk settings.



Node Status:
Shows the status of either a single host server or multiple servers clustered across multiple stateless nodes running in High Availability Mode.

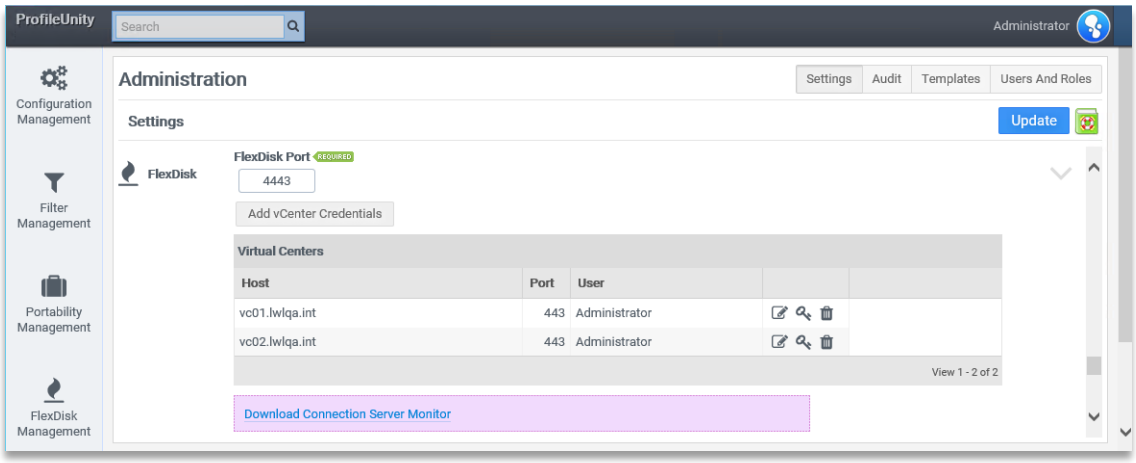
Enable High Availability Mode:
Check to enable High Availability Mode and then click on the blue **Update** button. Wait for ProfileUnity to switch into HA mode. You will see a message and then be redirected to the Management Console login when it is finished.

After logging in to the Management Console again, go to **Administration > Settings** scrolling to the **Clustering** section. Click on the **Add Node** link at the top right of the Node box and enter another node address. This address should be the fully qualified domain name of another server where an additional installation of the ProfileUnity Management Console was installed appended with the default port of 8000. For example, the node address would look like: prou2.mydomain.com:8000. Repeat the process to add more nodes.

The total number of nodes in the cluster must be an ODD NUMBER. We recommend 3 nodes, but 5 or 7 are supported as well. Generally more than 3 isn't necessary. The number of nodes that can go down and have the cluster still operating normally depends on the total number of nodes in the cluster. Normal operation requires (at least) 2 nodes to be up at all times. So a 3 node cluster would allow for a single node failure, a 5 node cluster would allow 3 nodes to fail and so on. Two nodes is not a valid cluster configuration and will NOT result in the ability to lose a node and still function normally!

FlexDisk

FlexDisk is a robust VMware VMDK (Virtual Machine Disk) delivery system managed by ProfileUnity. FlexDisk allows administrators to provision flexible, user VMDKs on a VMware Virtual Machine File system (VMFS) high-speed data storage volume to deliver user profiles or applications. FlexDisk settings go hand in hand with Clustering settings.



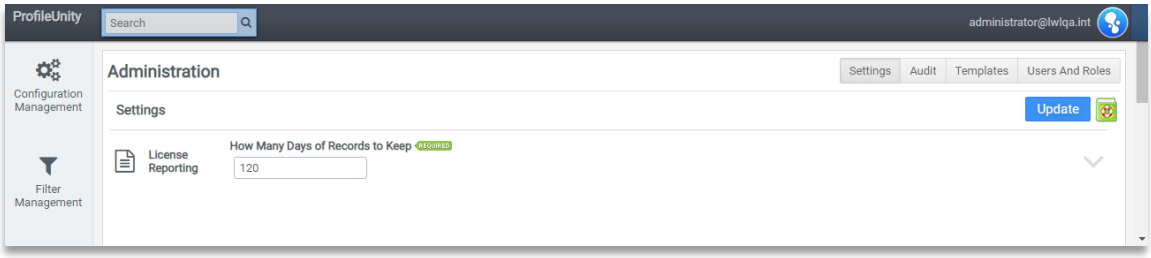
FlexDisk Port:
Enter the number of the port used to communicate with the FlexDisk API. The default is 4443.

Add vCenter Credentials:
Click this button to enter your VMware Virtual Center credentials. Enter your credentials formatted the same as they are in the Virtual Center Client. The FlexDisk technology supports the usage of multiple VMware vCenter Servers. For organizations that have users that float between multiple vCenter Servers, FlexDisk coordinates communication and executes administrative tasks using the vCenter APIs to attach and detach VMDKs based on each user's requirements and the server to which they are currently attached.

Virtual Centers:
Lists the VMware vCenter Servers that FlexDisk has access to. You can edit or remove servers from this list.

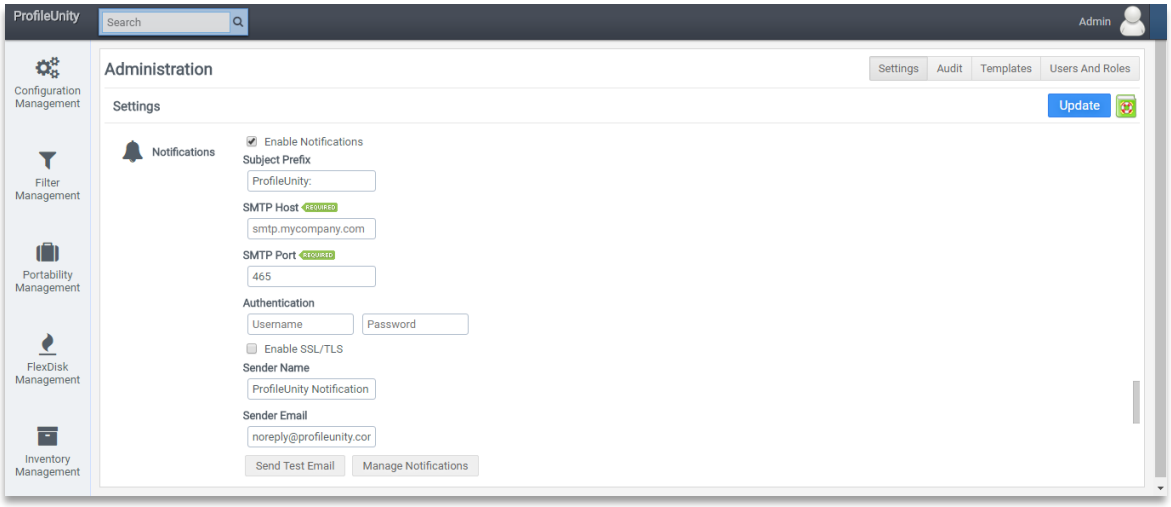
Download Connection Server Monitor:
Click to download the Connection Server Monitor. Install the CSmonitor service on all View Connection Brokers the users will be using to access pools. You will tell the CSmonitor about the primary node installation of the ProfileUnity Management Console.

License Reporting



How Many Days of Records to Keep:
Enter the number of days to retain user license information. The default is 120 days.

Notifications



Enable Notifications:
Check this option to enable email notifications from ProfileUnity.

Subject Prefix:
Enter the prefix that will be placed on the subject line of all ProfileUnity email notifications.

SMTP Host:
Enter the address of the SMTP server that will send the emails.

SMTP Port:
Enter the port number for the SMTP server to use for communication.

Authentication:
Enter the **Username** and **Password** that will be used for authentication purposes when using the SMTP server. The **Enable SSL/TLS** option is available to encrypt transmissions when sending emails.

Sender Name:
Enter the name that the email will be "From" when it is delivered.

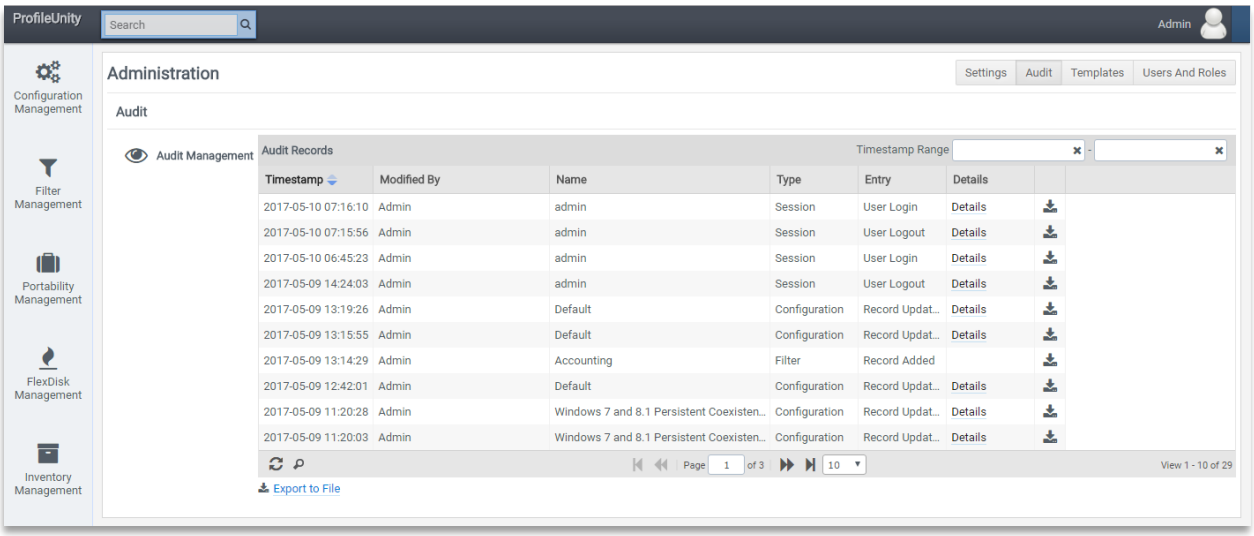
Sender Email:
Enter the email address that the email will be "From" when it is delivered.

Send Test Email:
Allows you to test your email notification settings by sending a test email from within ProfileUnity to an email address that you specify. Please click on the blue **Update** button to save any setting changes before sending a test email.

Manage Notifications:
Manage a list of users who will receive notification emails. ProfileUnity uses this list to email users about node outages when using the FlexDisk technology. Separate multiple emails in this list with a semi-colon.

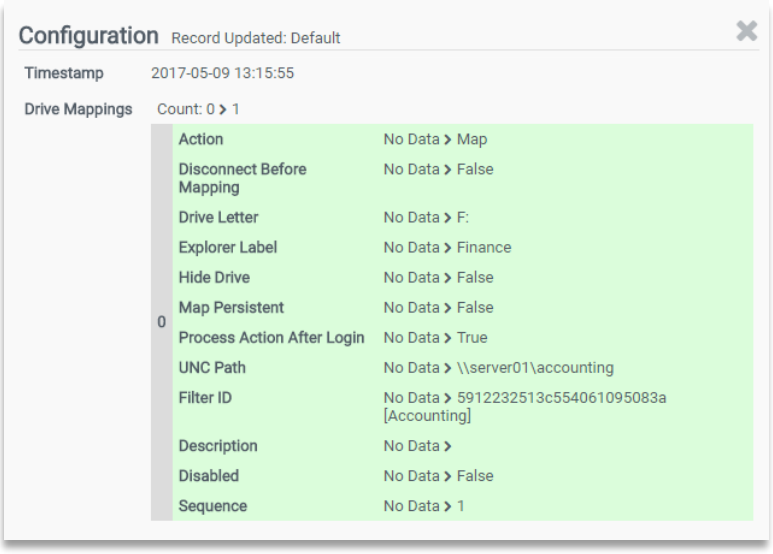
Viewing the Audit Trail

If Auditing is configured in the Administration Settings tab, ProfileUnity keeps track of changes made in your environment including logins to the Management Console, creation of new configurations, and edits made to filter, portability or configuration rules. Audit Management shows when the changes were made and who modified it. To get to these settings, go to your login user ID at the top right of the Management Console interface and select **Administration**. At the top of the Administration area, select the **Audit** tab.



Use the **Timestamp Range** fields at the top of the Audit table to limit your search based on a start and end date.

To view the details of each audit record inside of the ProfileUnity Management Console, click on the **Details** link. For each data field, the previous value and the new value are shown.



To export a CSV file of changes from each individual record, click on the **Download** icon at the end of each row.

To export a CSV file containing a comprehensive list of all changes, click on the **Export to File** link at the bottom of the Audit table.

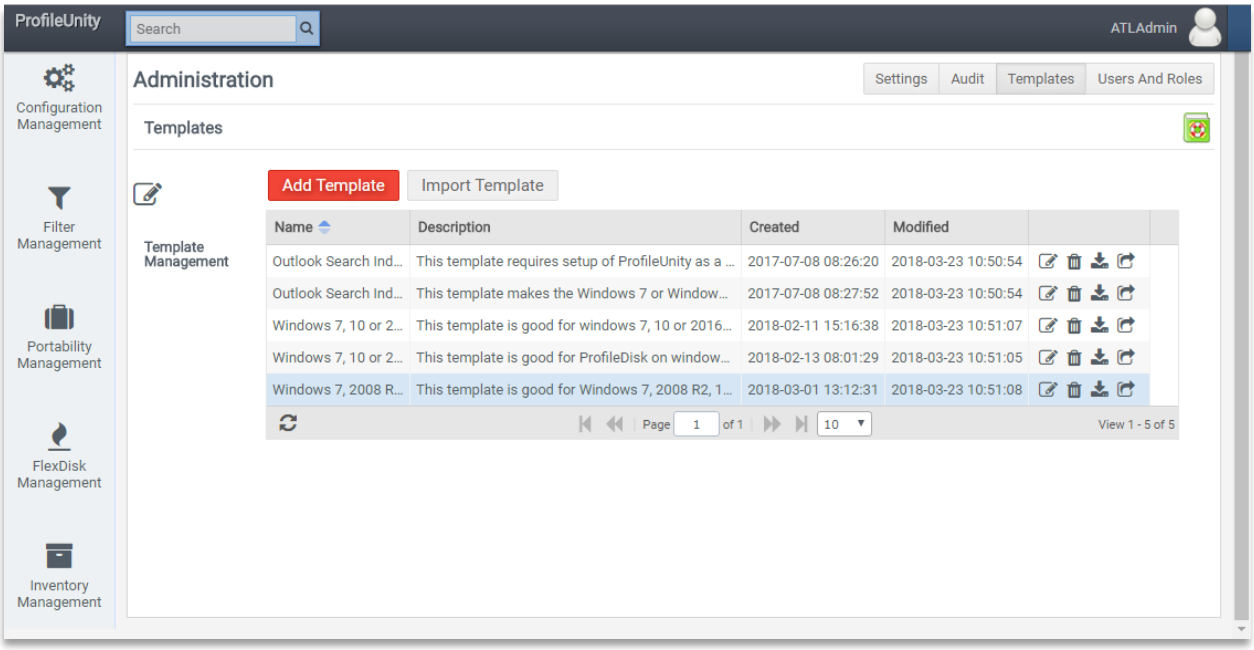
Please note that you may need to allow text wrapping in order to see all the information contained in the "Details" column in the exported CSV files.

To change how many days of records Audit Management should keep or to turn Audit Management on or off, please see the **Audit** section of the **Administration > Settings** tab.

Managing Configuration Templates

The Guided Configuration Wizard makes use of configuration templates to jump start the configuration process. The standard templates provided by Liquidware include about 85% of the Windows settings a user would need right out of the box. While ProfileUnity comes with several different useful templates, the Template Administration area allows console administrators the ability to create additional configuration templates or manage the existing templates they have. Configuration Templates can help administrators build configurations for their environment faster than creating each configuration module rule individually, saving valuable time. Instead of creating a new configuration from scratch, the administrator can choose a template as a basis for a new configuration while running the Guided Configuration Wizard. If the template has the majority of the required settings already configured, then the administrator only has to make minimal changes to create the new configuration which is specific to the enterprise environment.

To manage your configuration templates, go to your login user ID at the top right of the Management Console interface and select **Administration**. At the top of the Administration area, select **Templates**.



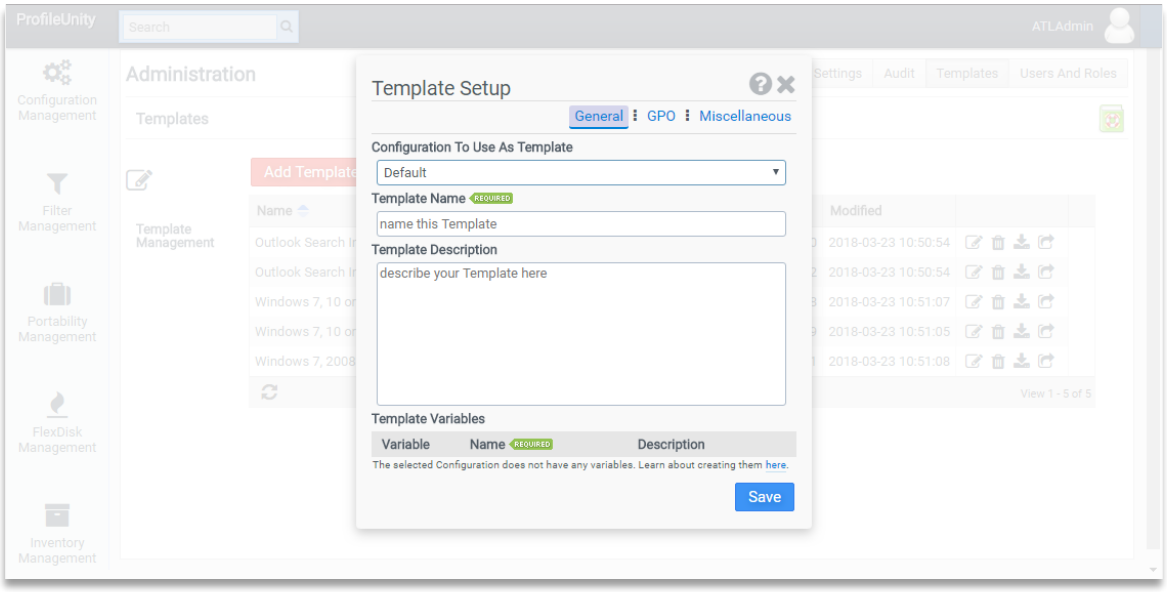
The Template list contains all of your existing configuration templates including the standard templates that come with ProfileUnity.

Importing Configuration Templates

If you have received a template from a partner or been working on perfecting a template with an outside resource, you can import it into the Management Console by clicking on **Import Template** and choosing the location of the JavaScript Object Notation (JSON) file. The configuration template will be added to the Template list and you will be able to select the new template the next time you run the Guided Configuration Wizard to create a new configuration.

Creating Configuration Templates

Any of the configurations you create in the Configuration Management area can be saved as templates. These templates can then be used over and over again by the Guided Configuration Wizard. Once you have tweaked the settings in your configuration to match your environment, you can add a new template by clicking on **Add Template** and completing the following information. Note there are three sections of settings for templates: General, GPO, and ProfileDisk. Click **Save** to create the new template.



Configuration to Use as Template:
Displays a list of all your current configurations from the Configuration Management list. Choose one that you would like to base this template off of.

Template Name:
Enter the new template name.

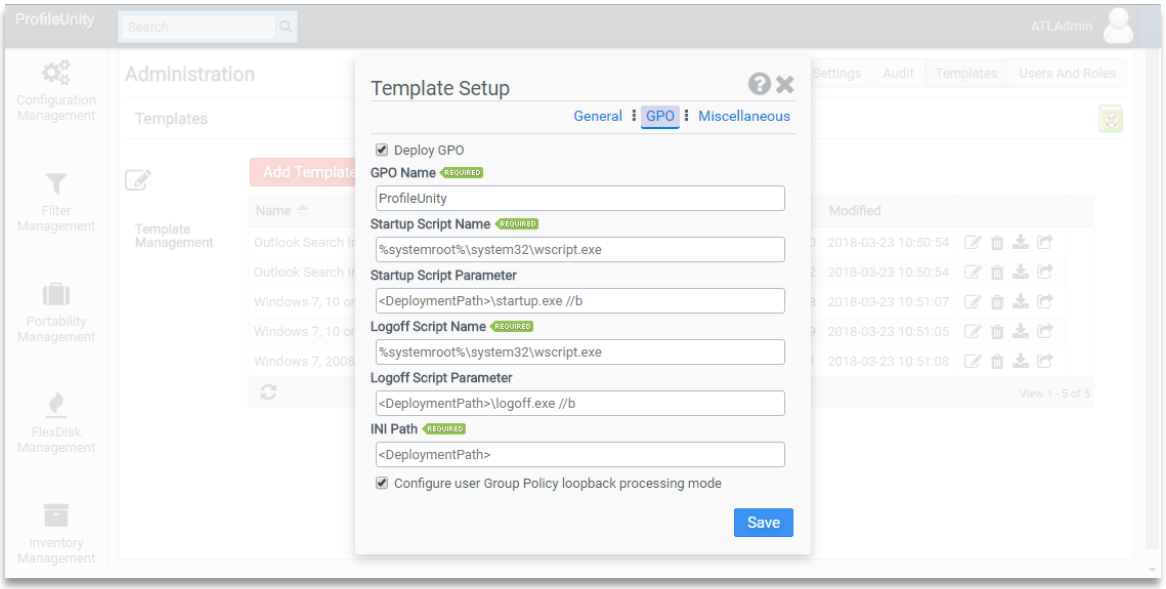
Template Description:
Enter the description of this template. This is optional but will be useful to the user setting up a new configuration based on this template.

Template Variables:
These are variables that are found within the selected Configuration. You must provide a name for each variable. Providing a description will help the user understand what the variable is meant for.

NOTE: Variables are marked in Configurations as #MyVariable#. Syntax must be exact. Use CamelCasing for variables if you would like ProfileUnity to automatically detect spaces when assigning names.

Good Examples:
#NetworkFolder#
#ProfileCleanupFilter#

Bad Examples:
#NetworkFolder (missing trailing #)
NetworkFolder# (missing initial #)
#Network Folder# (do not use spaces between #)



Deploy GPO:
Check this option to instruct the template wizard to deploy a GPO.

NOTE: The user creating the configuration must be part of the Active Directory domain.

GPO Name:
If **Deploy GPO** is checked, enter the name of the GPO. ProfileUnity will assign a UUID to the end of the GPO to ensure uniqueness.

Startup Script Name:
If **Deploy GPO** is checked, enter the path to the startup script.

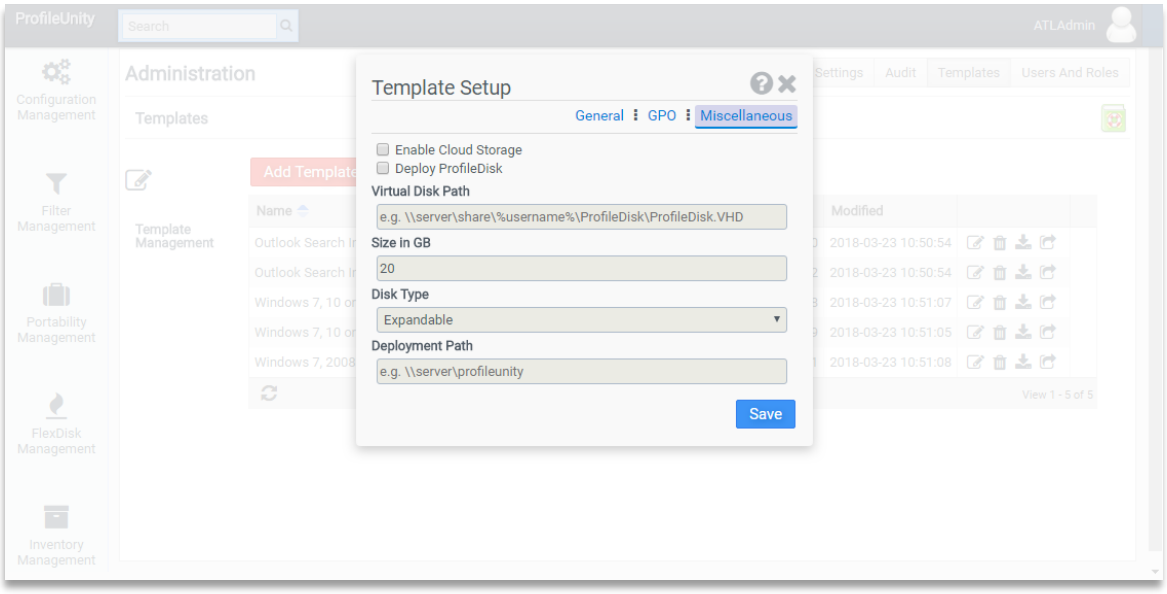
Startup Script Parameter:
If **Deploy GPO** is checked, add any parameters to pass to the startup script.

Logoff Script Name:
If **Deploy GPO** is checked, enter the path to the logoff script.

Logoff Script Parameter:
If **Deploy GPO** is checked, add any parameters to pass to the logoff script.

INI Path:
If **Deploy GPO** is checked, enter the location of the INI path file.

Configure user Group Policy loopback processing mode:
If **Deploy GPO** is checked, you may check this option to enable the Group Policy loopback policy when the GPO is deployed.



Enable Cloud Storage:
This instructs the template wizard to make use of cloud storage.

Deploy ProfileDisk:
This will instruct the template wizard to deploy a ProfileDisk configuration.

Virtual Disk Path:
Enter the location of your virtual disk.

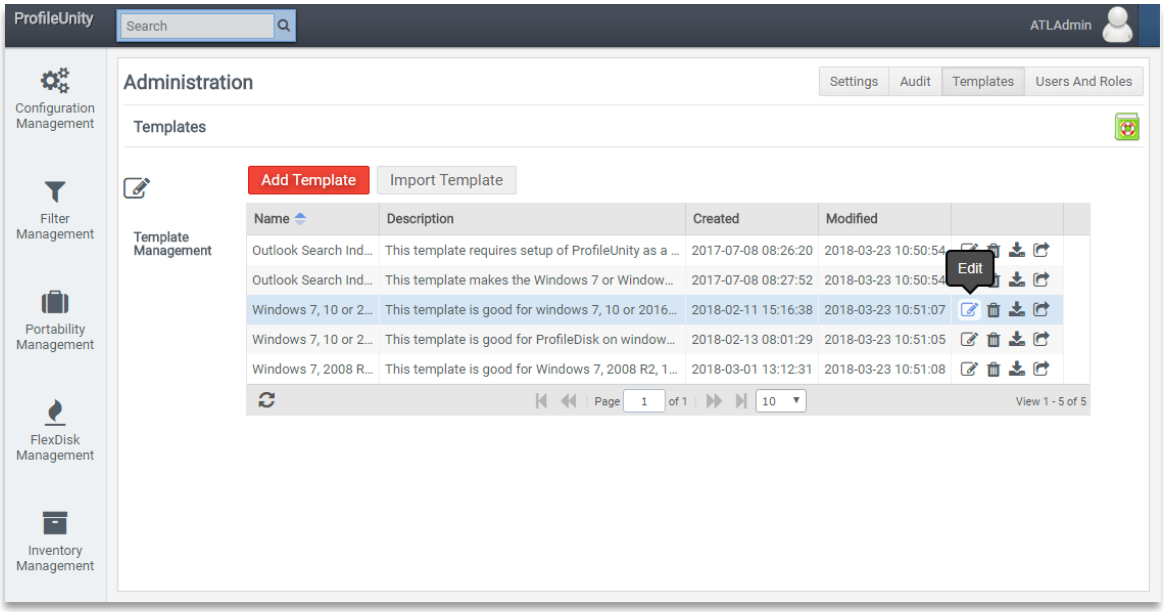
Size in GB:
Enter the maximum size of the virtual disk in gigabytes. Check the **Enable Compression** option to allow space savings.

Disk Type:
Choose whether this disk will be **Expandable** or **Fixed** in size.

Deployment Path:
Enter the deployment location where the ProfileDisk configuration will be stored on your NETLOGON share.

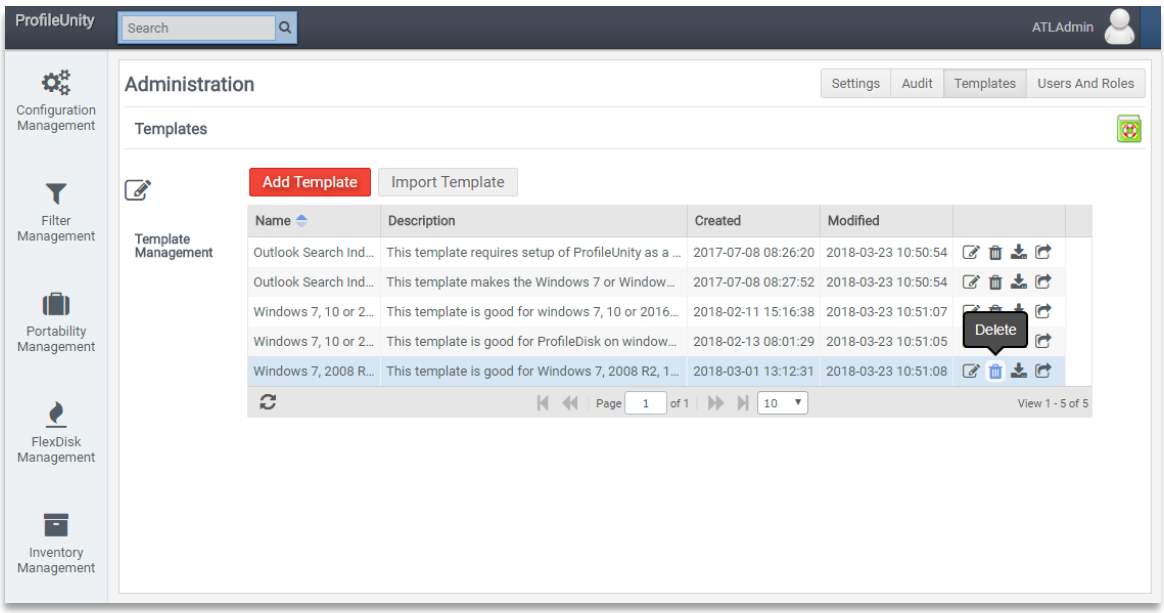
Editing Template Settings

To make changes to a template, choose one from the Template List and click on the **Edit** icon to the right of the template's name. Any of the settings previously noted under **Adding Configuration Templates** can be modified.



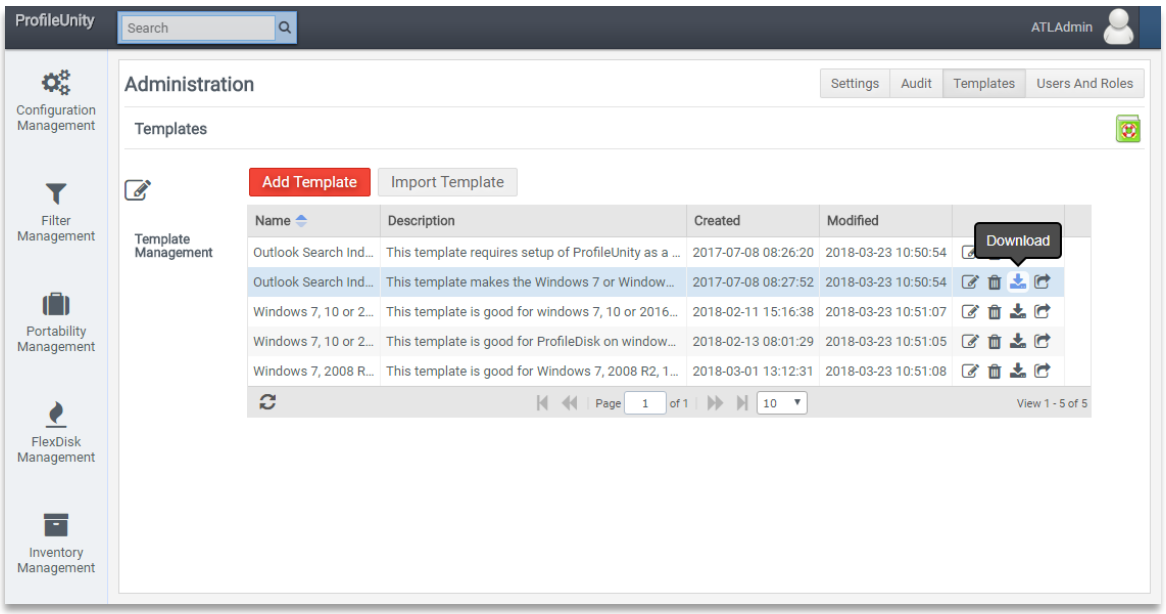
Deleting Templates

To delete a configuration template, choose the template from the Template List and click on the **Delete** icon to the right of the template's name. You will be asked to confirm the deletion. Please note that once the template is deleted, it cannot be recovered.



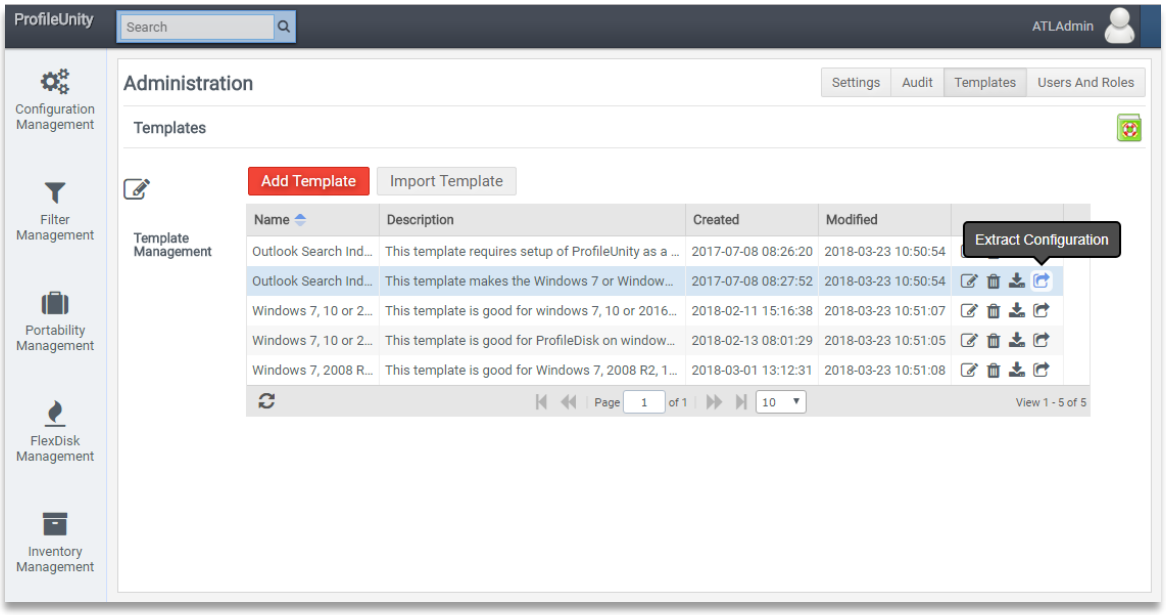
Exporting Templates

To export a particular configuration template, choose it from the Template list and click on the **Download** icon to the right of the template's name. Then choose the location where you would like to save the JSON template file.



Extracting Template Configuration Settings

To extract a template's settings, choose it from the Template list and click on the **Extract Configuration** icon to the right of the template's name. This action extracts the configuration settings from the template and adds it to the list of configurations in the Configuration Management section with all settings configured. Admins can easily extract template settings, quickly make changes in the configuration, and then add the edited configuration as a new template.

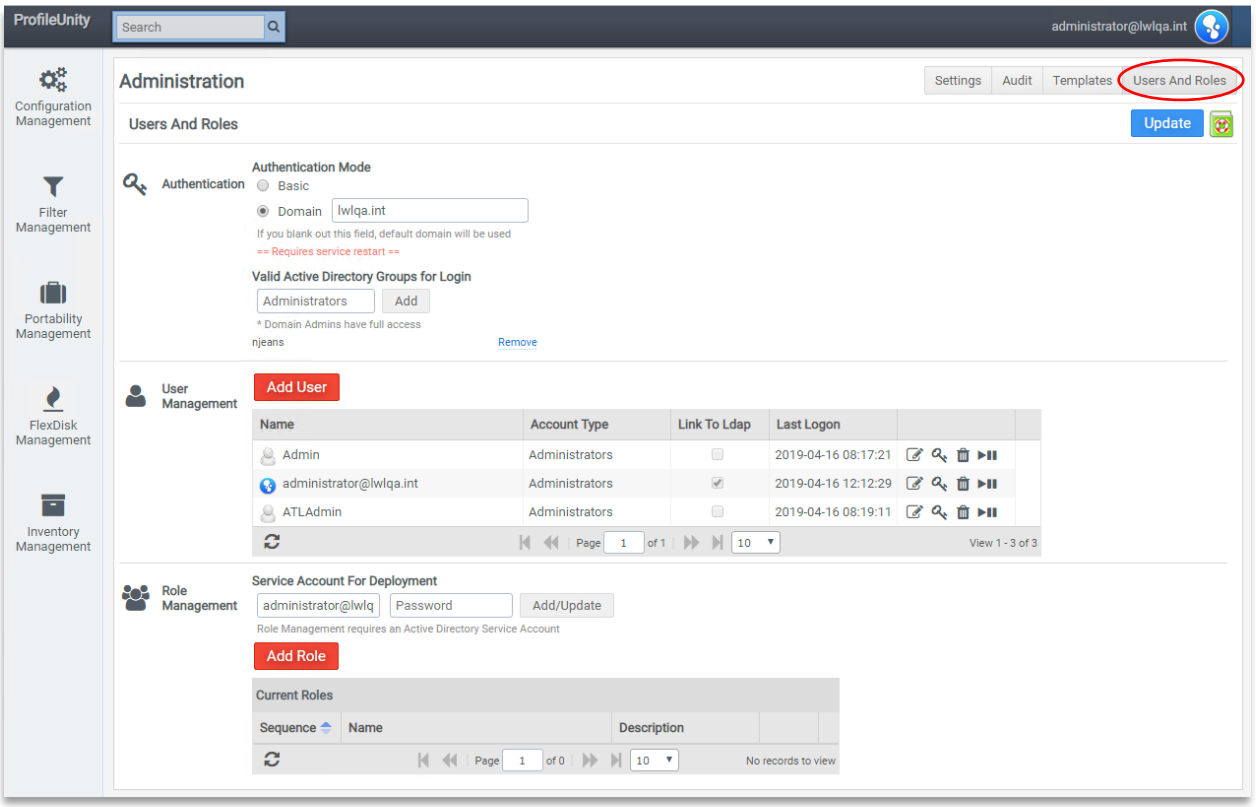


Managing Console User Accounts & Roles

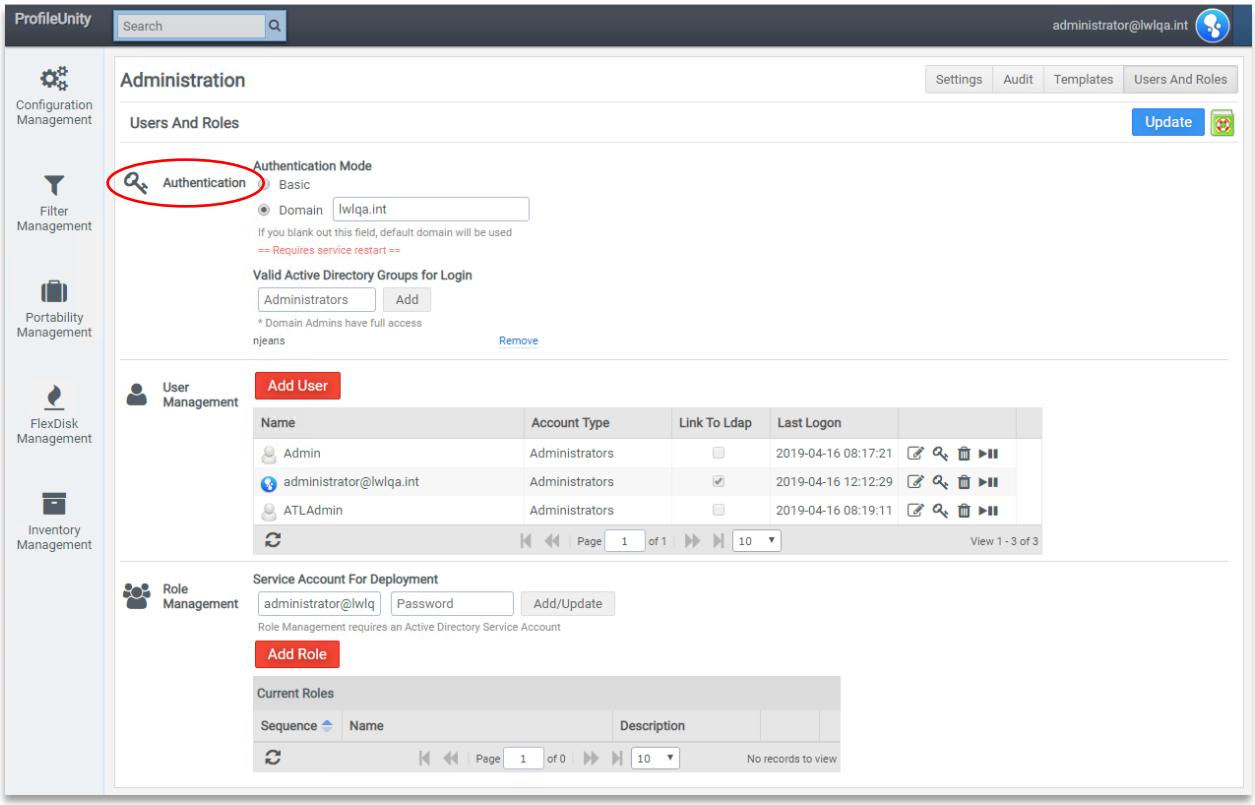
The ProfileUnity Management Console provides one central utility administrators can use to configure and manage how users interact with the different types of desktops in their environment. Upon installation, ProfileUnity creates a default admin user account. The username is “admin” and you are asked to supply a password. Please make note of this information to refer to it later as needed.

You may edit these settings and create additional console user accounts if you wish. Creating, modifying, and deleting user accounts is done through the User Administration area of the ProfileUnity Management Console. To get there, go to your login user ID at the top right of the interface and select **Administration** from the drop-down list.

Under **Administration Settings,** select **Users And Roles**.



Configuring Authentication Settings



Authentication Mode:

ProfileUnity offers two ways to authenticate users: **Basic** or **Domain**. Basic authentication uses ProfileUnity's local authentication process. Domain authentication allows users to authenticate using Active Directory during a ProfileUnity Management Console logon. This feature allows existing directory credentials to be used with ProfileUnity and prevents users from having to maintain an additional password.

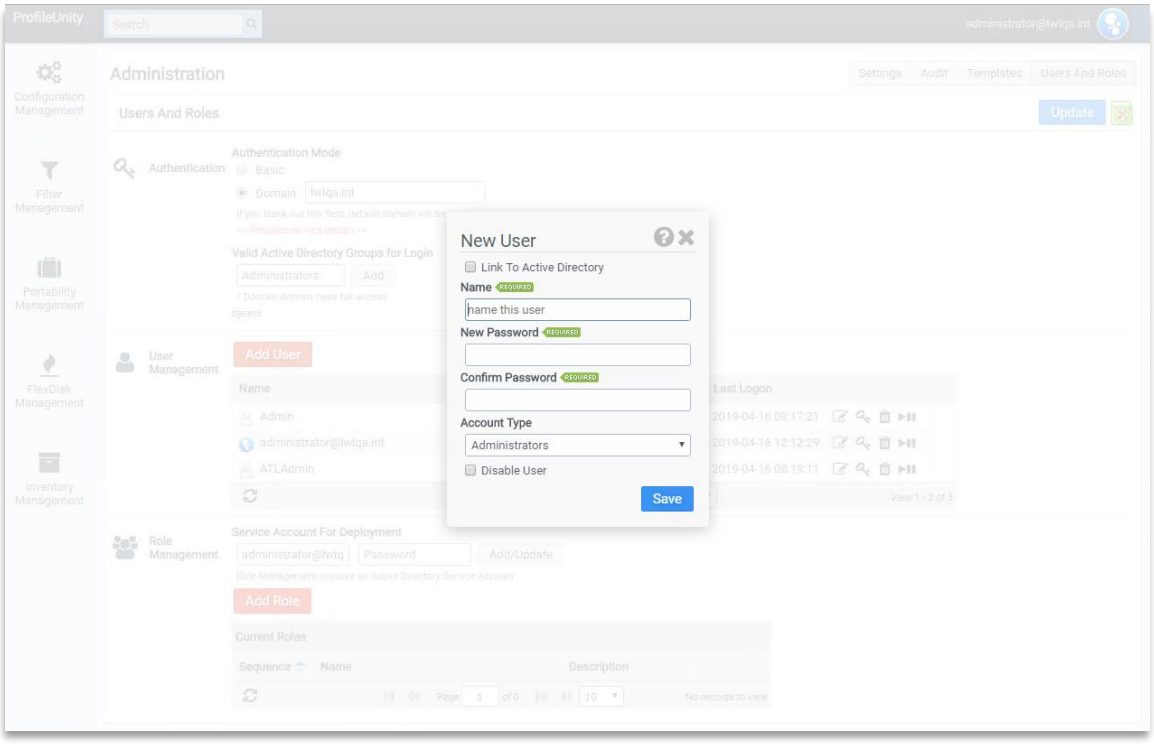
Any change to the **Authentication Mode** requires a service restart.

Valid Active Directory Groups for Login:

Allows users to enter Active Directory groups that are allowed to login to ProfileUnity. Simply type in the name of the group and click **Add**. To delete a group, click on **Remove** next to its name.

Creating a New User Account

To add a new console user, click on the **Add User** button in the User Management list.



Enter the following information for your new user:

1. Select **Link to Active Directory** to authenticate users with LDAP.
2. Enter the **Name** the user will use as their username to logon to ProfileUnity.

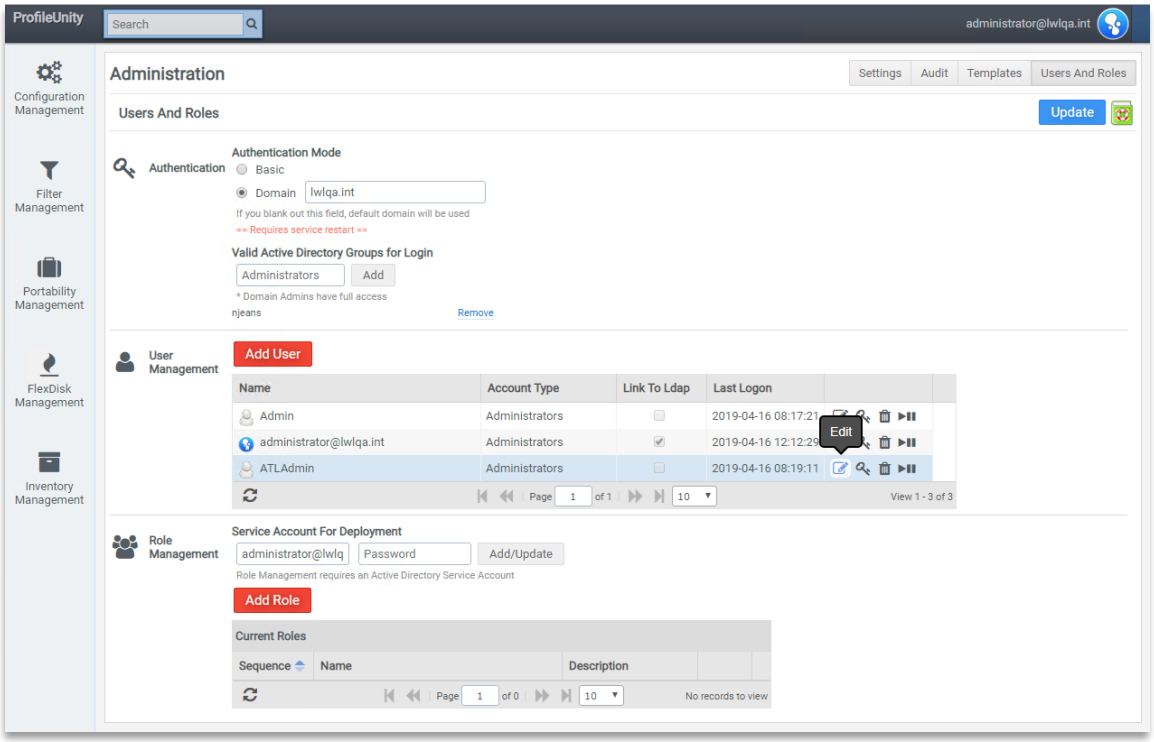
> ***NOTE: If Link to Active Directory is selected, this username must map to an entry in the LDAP directory server for authentication to succeed.***

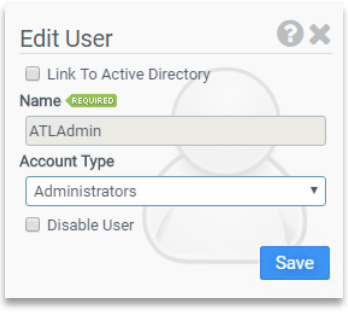
3. Enter the **New Password** the user will use to logon to ProfileUnity. This field is not used if **Link to Active Directory** is selected. A valid password must be at least 8 characters long and include at least one character from three of the following categories:
    a. Upper case letters
    b. Lower case letters
    c. Numbers
    d. Non-alphanumeric characters
4. Re-enter your password in the **Confirm Password** field. This field is not used if **Link to Active Directory** is selected.

5. Select an **Account Type:**
   - **Administrators**
   - **Users**
6. Selecting **Disable User** prevents the user from logging in to the ProfileUnity Management Console.
7. Click **Save** to save all your changes before leaving the User Editor. All unsaved changes will be discarded.

Editing User Account Settings

To edit a console user's settings, click on the **Edit** button next to the user's account name in the User Management list.





When editing user settings, modify the following options if necessary:

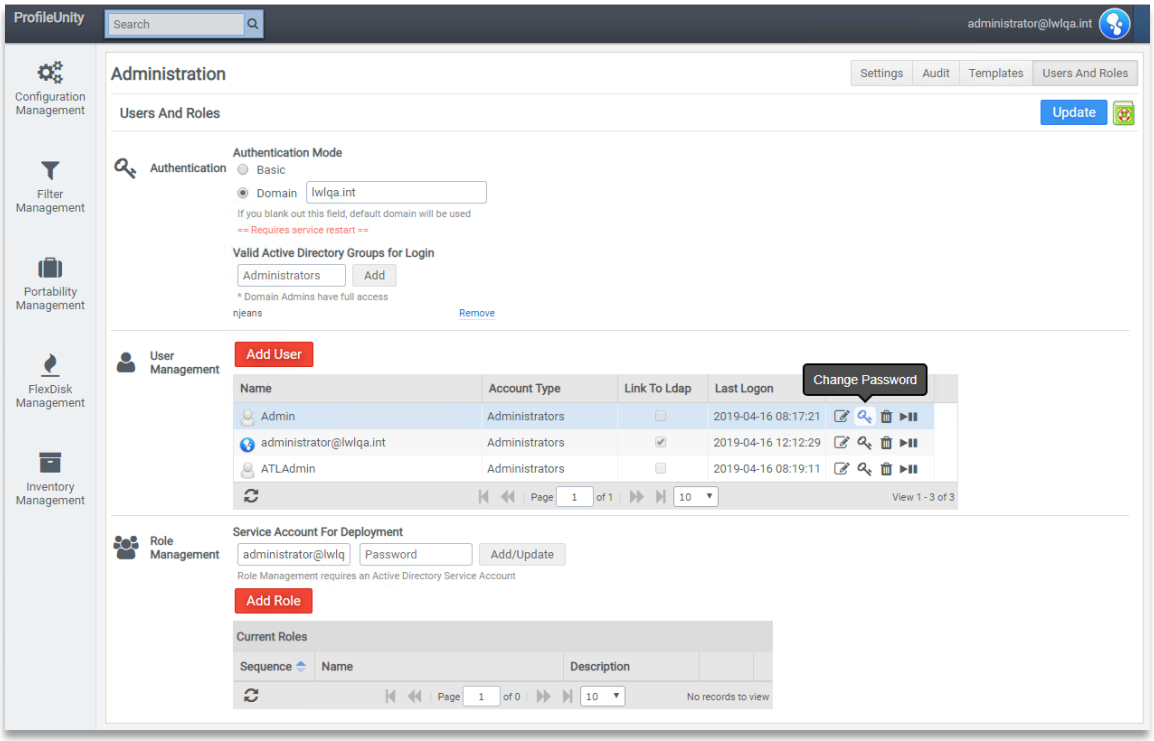
1. Select **Link to Active Directory** to authenticate users with LDAP.
2. Enter the **Name** the user will use as their username to logon to ProfileUnity.

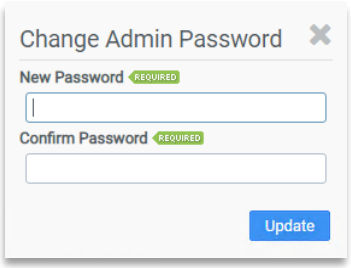
NOTE: If Link to Active Directory is selected, this username must map to an entry in the LDAP directory server for authentication to succeed.

3. Select an **Account Type:**
   - **Administrators**
   - **Users**
4. Selecting **Disable User** prevents the user from logging in to the ProfileUnity Management Console.
5. Click **Save** to save all your changes before leaving the User Editor. All unsaved changes will be discarded.

Resetting User Account Passwords

To reset an existing console user's password at any time, click on the **Change Password** icon next to the user's account name.



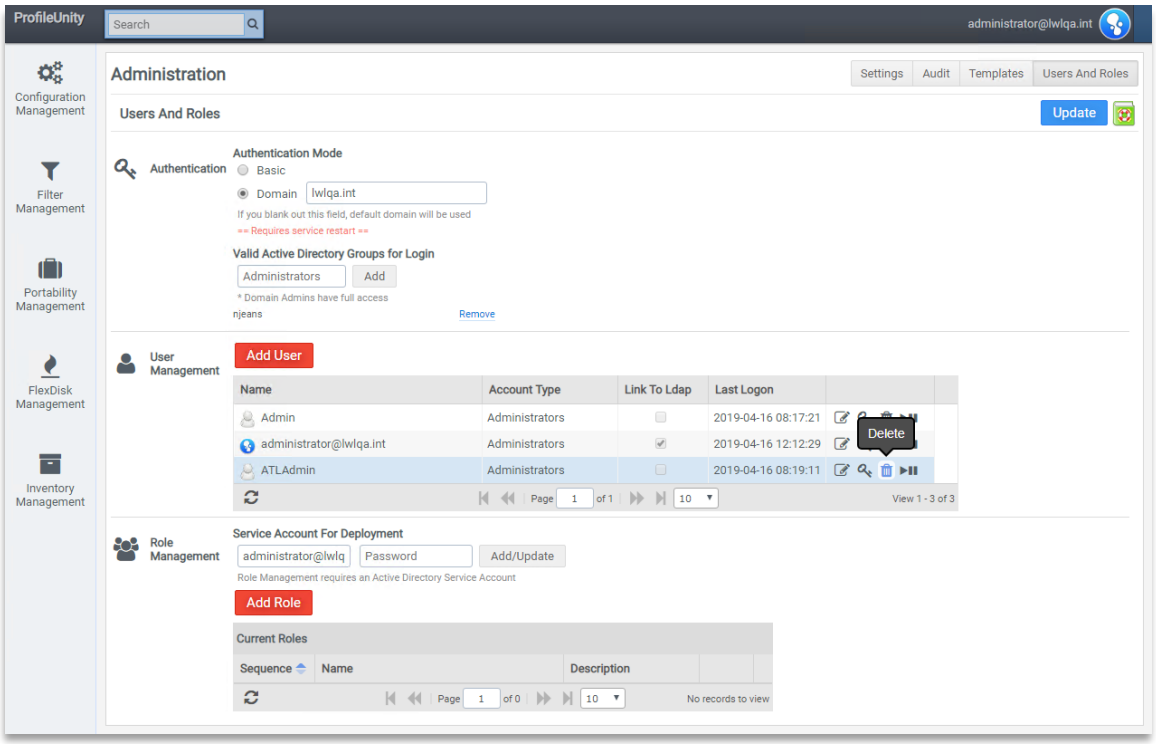


You will be asked to enter a new password and to confirm the password by re-entering it. A valid password must be at least 8 characters long and include at least one character from three of the following categories:

- Upper case letters
- Lower case letters
- Numbers
- Non-alphanumeric characters

Deleting User Accounts

Removing old or unused user accounts is easily done from the User Management list. Simply select **Delete** next to the name of the user you want to remove. As a safeguard, ProfileUnity will ask you to confirm the deletion. Please note that once the user account is deleted, it cannot be recovered.



Disabling or Enabling User Accounts

Disabling active user accounts user accounts can be done in two ways. You can either choose to **Edit** the user Account and check the **Disable User** option or you can just select the **Enable or Disable** button in the User Management list next to the name of the user whose account you want to disable.

To reactivate inactive user accounts, simply click on the **Enable or Disable** button in the User Management list next to the name of the user whose account you wish to enable again.

Disable Toggle – Account is **Enabled**. Click the dark gray icon to disable the user account.

Enable Toggle – Account is **Disabled**. Click the grayed-out icon to enable the user account.

Role Management

Role management offers the ability to assign specific management console editing permissions for each Active Directory user account. This feature allows there to be leveled roles within a help desk team that allow certain configuration changes to be made by some while reserving other edits for higher level admins. For example, a level-one role might allow changes to drive mapping, printer, and shortcut settings but not have permission to change filters, portability, or FlexDisk settings. By default, only local admin users can make edits in the ProfileUnity Management Console. All other AD users can view the current settings, but are unable to make edits until roles are assigned. Roles open up access to grant granular editing permissions within the ProfileUnity Management Console. However, ProfileUnity cannot grant or override server user account permissions.

Users do not have to be Active Directory Administrators in order to be assigned roles, but their accounts need to be able to query AD and write to one or more deployment paths.

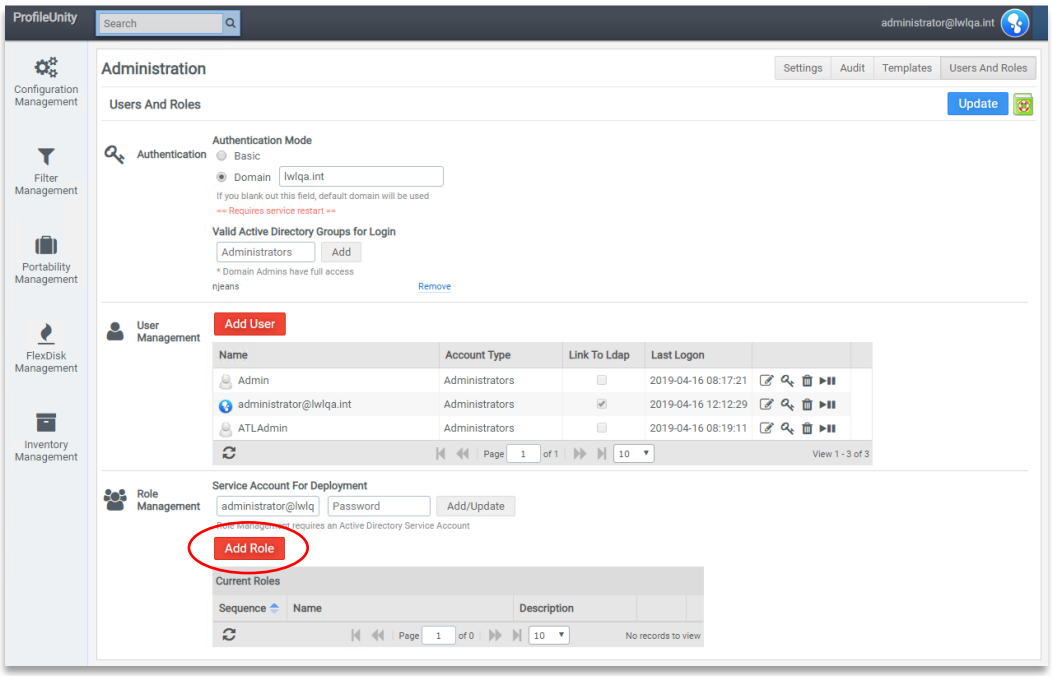
Designating a Service Account

One Active Directory account must be designated as the account that will deploy the configuration file and make queries to outside resources like AD, file shares and print shares. This account will need full control to the deployment path. Users do not need any access to the deployment path when a service account is setup. Please enter the username and password for this account. Then click the **Add/Update** button.

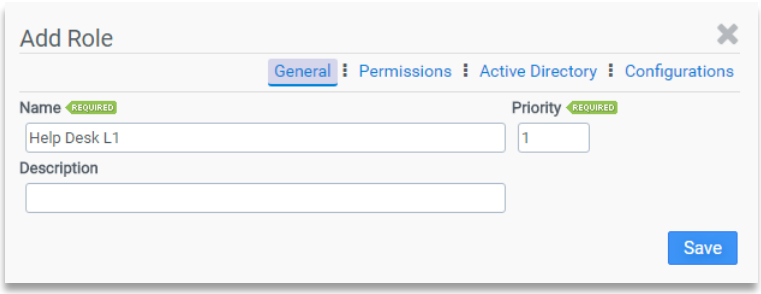
Creating a New Role

To create a new role,

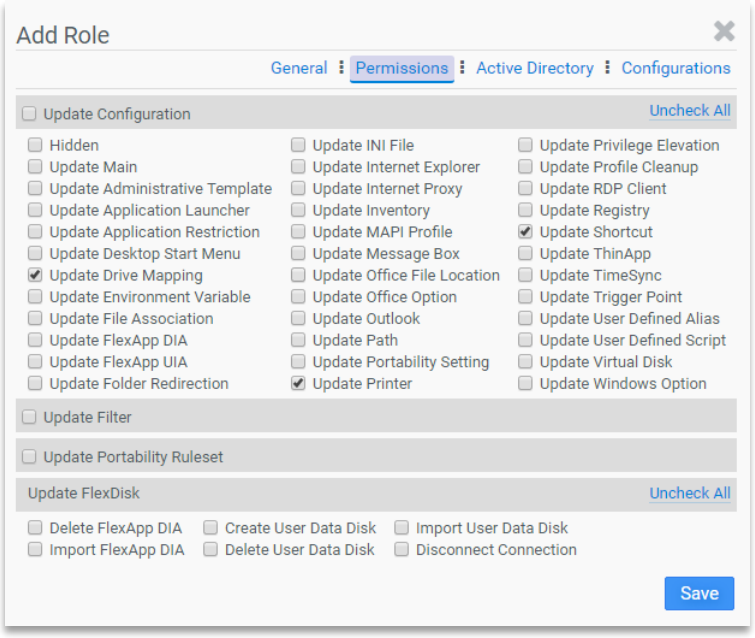
1. Click on the **Add Role** button.



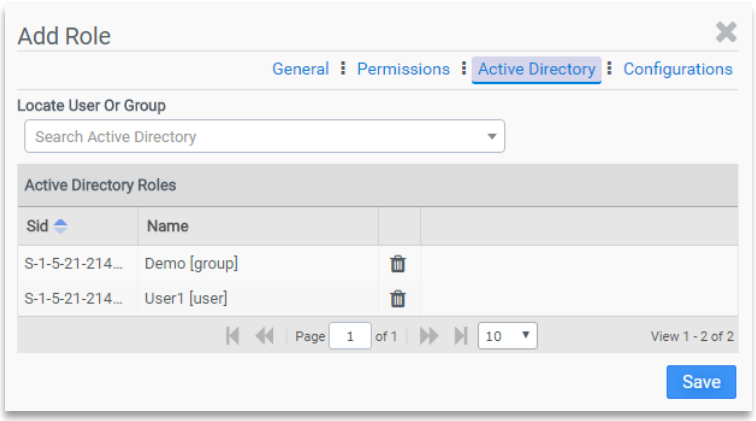
2. For the General section, type the **Name** of the Role and give it a **Priority**. If conflicting roles are created, the Priority determines which role will take precedence. The lower number priority wins. Type in an optional **Description**.



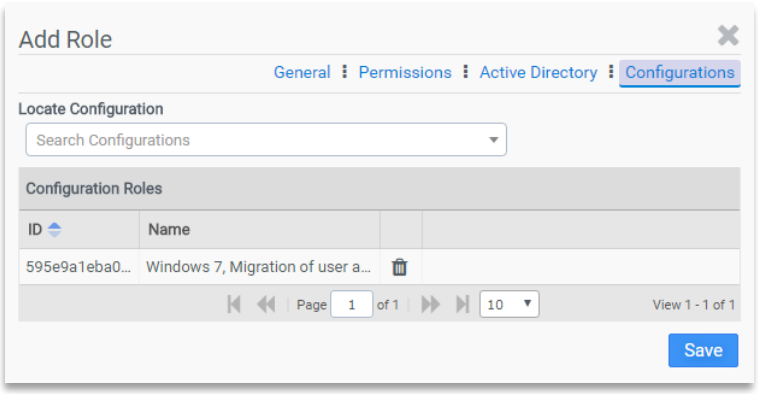
3. Advance to the Permissions section and select all permissions to grant to this new role.



4. Go to the Active Directory section and start typing the name of the user or group to be assigned to this role in the **Locate User or Group** field. Once you type in the first three letters, ProfileUnity will search for matching users or groups. Add as many users and/or groups to this list.



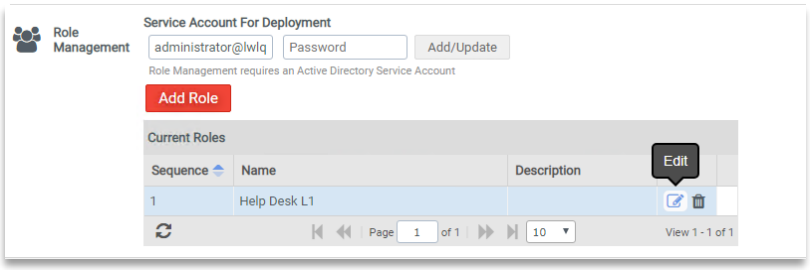
5. Proceed to the Configurations section and type the name of the ProfileUnity Configuration where this role will apply in **Locate Configuration**. Add more configurations if desired.



6. Once you have completed all sections, click **Save**.

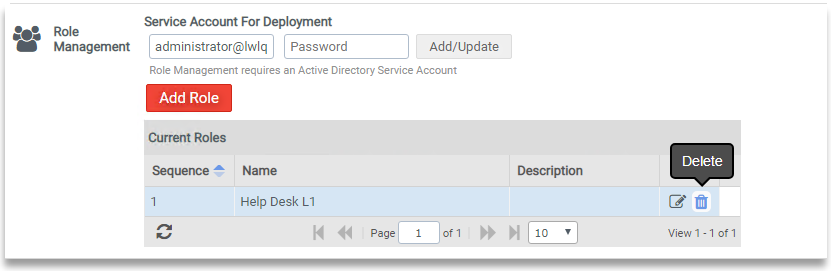
Editing a Role

To edit a role, click on the edit button to the right of the role name in the Role Management list. Make the necessary changes to each section as you did when creating a new role.



Deleting a Role

To delete a role, click on the delete button to the right of the role name in the Role Management list.



License Management

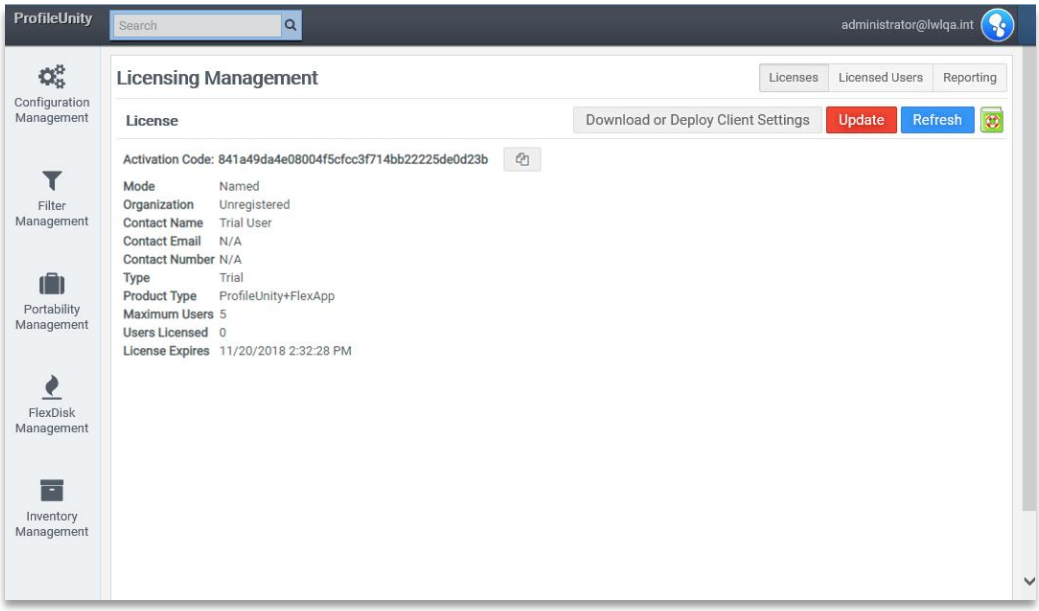
ProfileUnity with FlexApp v6.8.0 introduced a new licensing service that is built into the ProfileUnity Management Console. The new licensing service provides improved license management and greater flexibility with real-time usage in your environment. Because the ProfileUnity Management Console is processing license requests in real-time, the console should remain up and running at all times.

By default, ProfileUnity is designed to operate in evaluation mode for up to 5 users and up to 15 days. Once the evaluation period expires, administrators will no longer be able to download any changes to their Configurations. To request an extended evaluation period or to purchase a software license, please contact our Liquidware Sales Team.

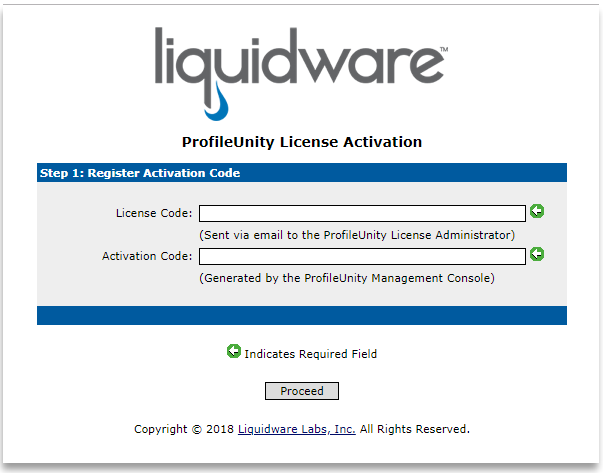
Activating or Updating Your Product License

ProfileUnity requires an activation process that generates a software license based on the combination of an Activation Code and a License Code. The two codes are combined in a secure Activation Portal and a unique license is generated for your ProfileUnity Management Console. License Codes are provided via email and sent to your organization's ProfileUnity Administrator. Your Activation Code can be found in the ProfileUnity Management Console after installation. To activate your license:

1. In the ProfileUnity Management Console, navigate to Licensing Management by hovering over your user name at the top right of the Management Console and select **Licensing** from the drop-down list.



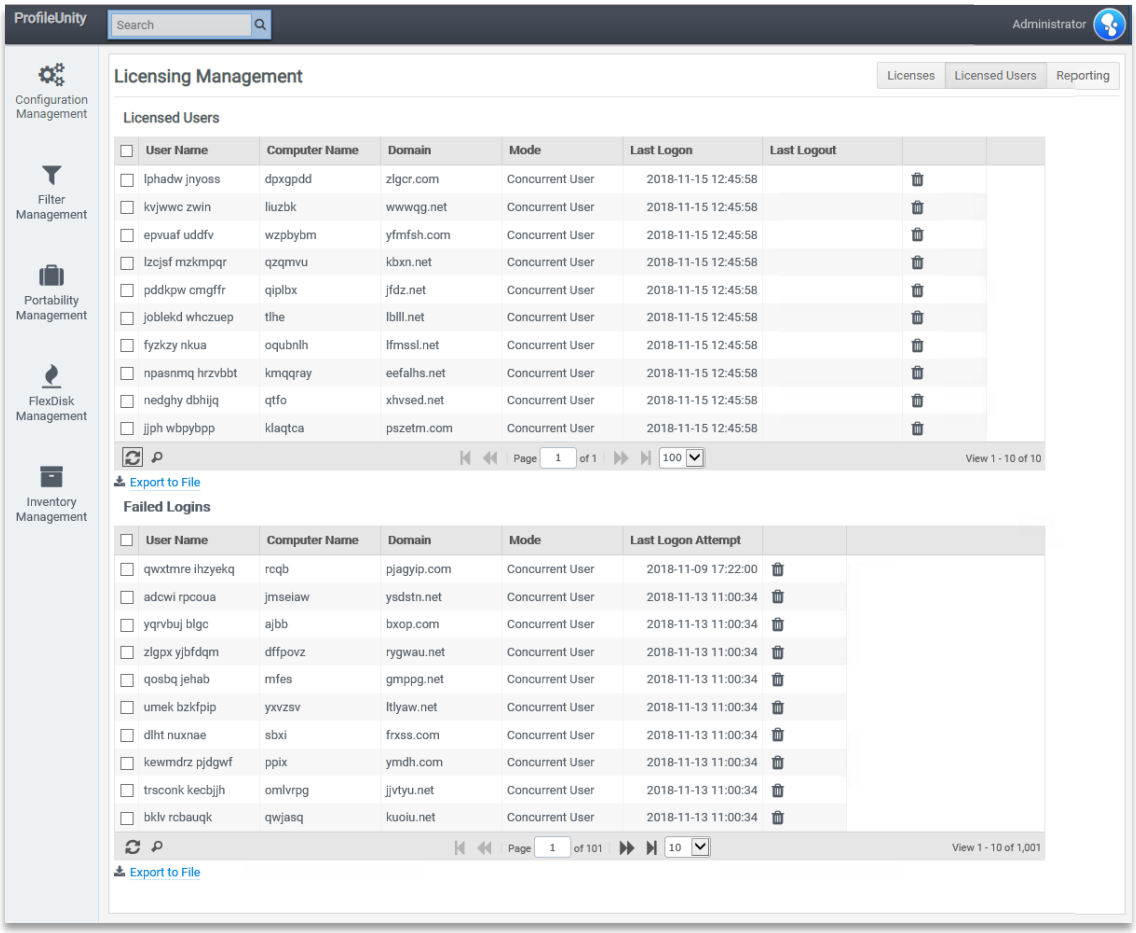
2. In a separate browser window, go to the ProfileUnity Activation Portal.



   a. Enter your **License Code** which can be found in the ProfileUnity License email that was sent to you from Liquidware.
   b. Enter your **Activation Code** which can be found in the ProfileUnity Management Console on the Licenses tab of the License Management area.
   c. Click the **Proceed** button in the activation portal and copy the generated license text.
3. In the Licensing Management area of the console, click the red **Update** button.
4. Paste the license text into the space provided.
5. Click on the blue **Import** button.
6. Click on the gray **Download or Deploy Client Settings** button to copy an updated clientsettings.xml to the ProfileUnity Deployment Path. This file is used by the ProfileUnity Client as end points are requesting a license when connecting back to the ProfileUnity Management Console.

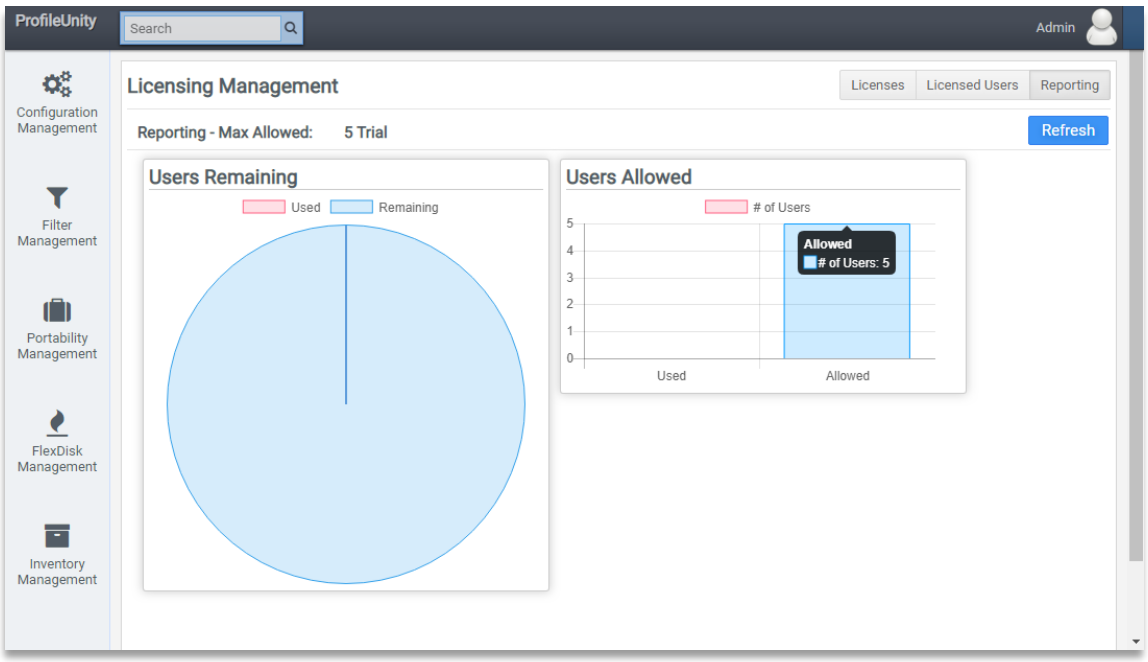
Managing User Licenses

The Licensed Users tab contains two tables. The first table displays users that are currently licensed for ProfileUnity. The second table displays a list of users whose license requests failed at login due to the number of users requesting licenses exceeding the license count. Administrators have the option to remove users from these searchable, sortable lists in order to free up licenses. The **Export to file** option allows you to export the users in that list to a CSV file.



Reporting

The Reporting tab in License Management allows administrators to see how many licenses they own vs. how many licenses are currently being used.



FlexDisk Technology Overview

FlexDisk is a robust VMware VMDK (Virtual Machine Disk) delivery system managed by ProfileUnity. ProfileUnity already provides application and data layering services while managing a user's personal settings across Windows devices. However, our FlexDisk technology offers even better application performance while significantly reducing storage, network, and CPU overhead. FlexDisk allows administrators to provision flexible, user VMDKs on a VMware Virtual Machine File system (VMFS) high-speed data storage volume to deliver user profiles or applications. These VMDKs are sharable among hundreds of users, reducing application and data storage space requirements. At login, FlexDisk instantly connects applications and data directly to floating pool users and avoids slow network streaming of applications and data, because it is hot-added from the VMFS to virtual machines when provisioned by View resulting in faster logins and better performance.

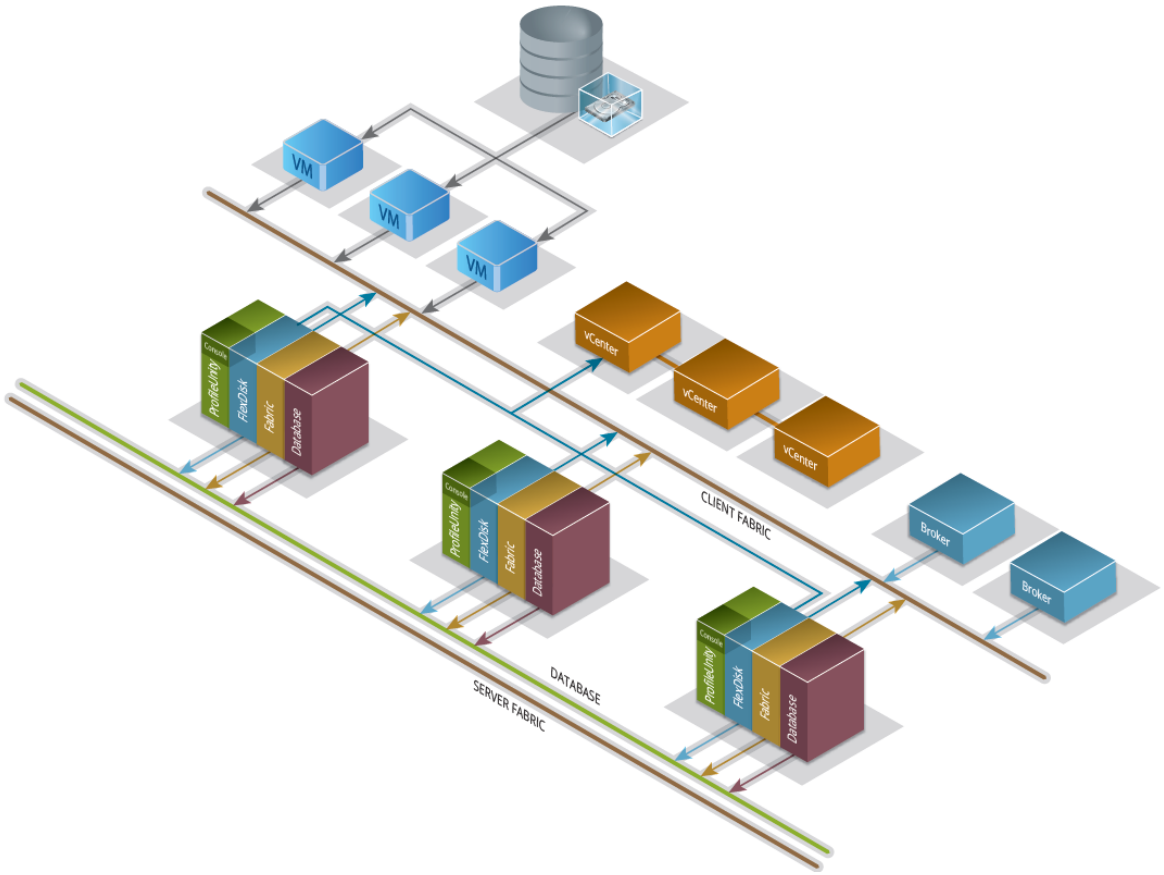
How might you take advantage of Liquidware's FlexDisk Technology? FlexDisk can be used to provide I/O steering, a follow-me user data disk, a follow-me user profile disk, and application layering. FlexDisk allows not only user applications to be on different VMDKs but also allows the OS to be separated from other user applications. Distributing the OS and applications across multiple VMFS volumes reduces the read IOPS on your base replica disk to get you more out of your storage. A follow-me user data disk is a read/write VMDK per user that follows the user from desktop to desktop. ProfileUnity will automatically detect the assignment of a user to a desktop and then hot-add their personal user data disk to that desktop. This follow-me disk allows for a location to store parts of the user profile, Outlook OST file, or even the My Documents folder on high-speed storage. In similar fashion, a follow-me user profile disk is where the user's entire profile exists on a VMDK that follows them from desktop to desktop. In this case, ProfileUnity attaches a VMDK that contains the user profile which removes the need to move the user profile and instead requires only a pointer to it.

FlexDisk also allows for VMDKs to be used to deliver FlexApp application layers for high performance execution of applications to non-persistent or persistent VMware View Desktops. Segmenting applications into different layers allows administrators to update application layers individually and push out layers to selective groups of users. Using this technique, each application layer is stored in its own VMDK and each user who is assigned to this application layer is given their individual write cache using a snap shot on the VMFS storage volume. Since the application is running from a VMDK on the same or similar storage as the VDI session, application performance will be very similar to an application baked into the base OS. With the application layer leveraging snap shots as the temporary write cache, it creates a fresh clean application layer each and every time. The applications can then be managed outside the base image and managed on a layer-by-layer basis.

FlexDisk Architecture

When a user's desktop depends on the delivery of a VMDK for the user's profile or application layers, it is critical to have a highly scalable and highly available system that can protect against failure. FlexDisk isn't just a single deliverable, but is a technology solution based on ProfileUnity Clustering where ProfileUnity is configured in a clustered mode to provide multiple nodes for scaling additional resources and to protect against a single point of failure. ProfileUnity Clustering is comprised of the following parts:

- Fabric (RabbitMQ)
- FlexDisk Service
- Connection Server Monitor (Broker Monitor)
- Database (MongoDB)
- ProfileUnity



Fabric (RabbitMQ)

The Fabric is a set of services used to get information from point A to point B without a single point of failure. The Fabric is used to transmit and receive login/logoff events so VMDKs can be added or removed from the desktop at the proper times. ProfileUnity is using RabbitMQ to serve as the Fabric. RabbitMQ is an efficient, highly scalable, open source messaging broker that enables applications to communicate with one another. RabbitMQ offers highly available messaging queues that can be replicated over several machines ensuring that all application messages are safe in the event of a hardware failure.

The Fabric helps FlexDisk avoid having mission critical actions going to a single IP that could fail or be over burdened with traffic. By using a messaging fabric, message data is replicated on many stateless nodes and multiple paths to get from point A to point B. In the event of a failure of any one node, the data can still continue to travel to point B via the other many stateless nodes and paths that are still available, creating a highly available and scalable system. Having many nodes ready to handle inbound data also creates a system that can scale when necessary, such as when many users are logging in simultaneously at the beginning of the workday. Whether you have a few hundred users or you have a few thousand users, the Fabric can dynamically adjust the number of nodes for scaling by adding or deleting nodes as necessary.

FlexDisk Service

The FlexDisk Service is the main service that is responsible for pulling needed information from the Fabric and handling all provisioning of VMDKs with VMware Virtual Center APIs. This includes creating VMDKs, formatting VMDKs, taking snap shots, and handling who gets which VMDK among other tasks. It's able to do all of its management remotely leveraging VMware vCenter APIs behind the scenes. When a user logs

into a desktop, a message is sent to the Fabric about the user and which desktop is logging in. Then one of the many FlexDisk services nodes processes the request and attaches the correct VMDK based on the administrator configuration.

Connection Server Monitor

This light weight broker agent sits on each View Broker and sends information to the Fabric about which user has logged in and logged off a desktop. Then one of the FlexDisk services nodes picks up the request to either hot-add the VMDK for a login or remove the VMDK for a logoff event.

Database (MongoDB)

ProfileUnity and its FlexDisk technology share a highly scalable and highly available database called MongoDB. MongoDB is an open source document database that stores data in the form of documents or JSON objects. Because the related data is stored together in documents, queries are much faster than in traditional relational databases where data from multiple tables needs to be joined together. MongoDB also provides a platform where data is distributed over replicated servers, providing enhanced performance and high availability while allowing the opportunity to easily horizontally scale the database. This database can be configured with a replica set of three or more nodes for high availability. This is done automatically for you when you make ProfileUnity aware of new nodes. The goal is to provide a layer of abstraction by automating MongoDB clustering and management as well as removing the need to have a database administrator just to manage the database.

ProfileUnity

ProfileUnity provides automation of deployment, management and integration of FlexDisk into the ProfileUnity UI to allow the delivery of a user profile or application layers via VMDK to persistent and non-persistent desktops. ProfileUnity keeps track of the status of each node's services in the Service Status Dashboard which can be found in the FlexDisk Integration section of the Administration area of the ProfileUnity Management Console. The consolidation of all services in the FlexDisk, Fabric (RabbitMQ) and Database (MongoDB) stack ensures ease of use for deployment, configuration and management of clustering by providing a layer of abstraction between ProfileUnity and the use of VMware VMDKs, removing the requirement for a complex setup, network load balancers, or external database clusters.

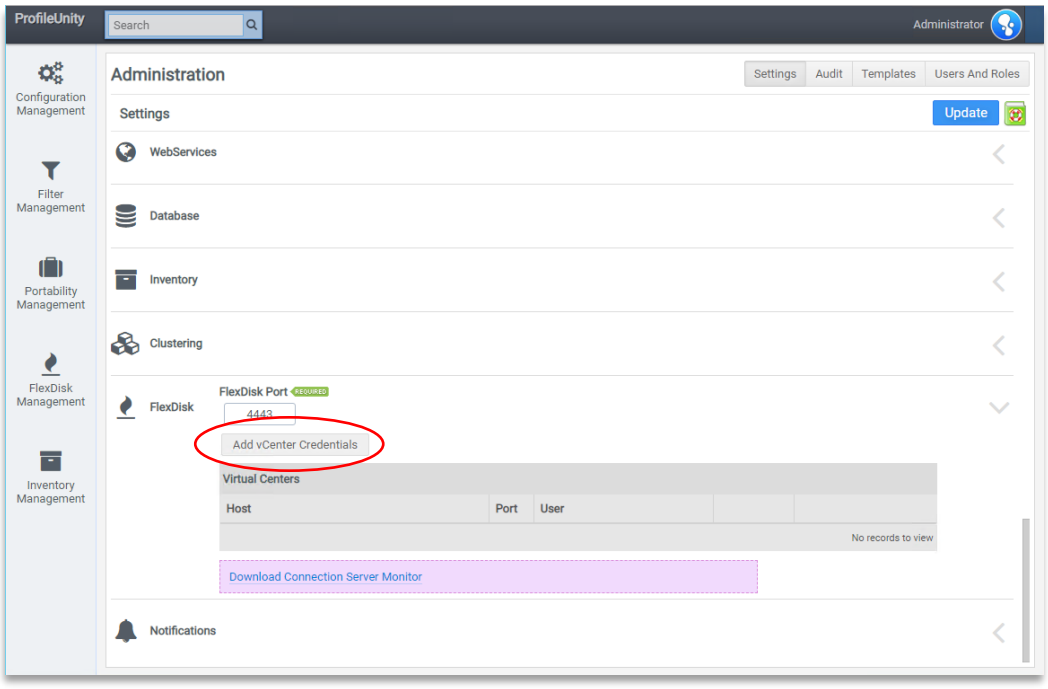
How to Setup ProfileUnity Clustering for FlexDisk

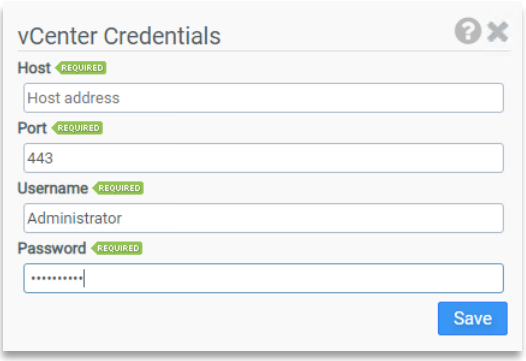
When setting up the FlexDisk service using ProfileUnity Clustering, following the instructions in order is very important. Each step builds the foundation for the next step and ensures all the pieces are communicating with each other. Note you must start with a minimum of three nodes, and the total number of nodes should always be an odd number if you wish to add more nodes.

Setting up the primary node:

1. Install the ProfileUnity Management Console on the first server. This machine must:
    a. Be running Windows Server 2008 R2, 2012 R2 or 2016,
    b. Have a static IP address, and
    c. Be joined to a domain.
2. Login to the ProfileUnity Management Console and go to the Administration Area by going to your user name at the top right of the Management Console and selecting **Administration** from the drop-down list.
3. On the Settings tab, scroll down to the FlexDisk section and click on the **Download Connection Server Monitor** link.

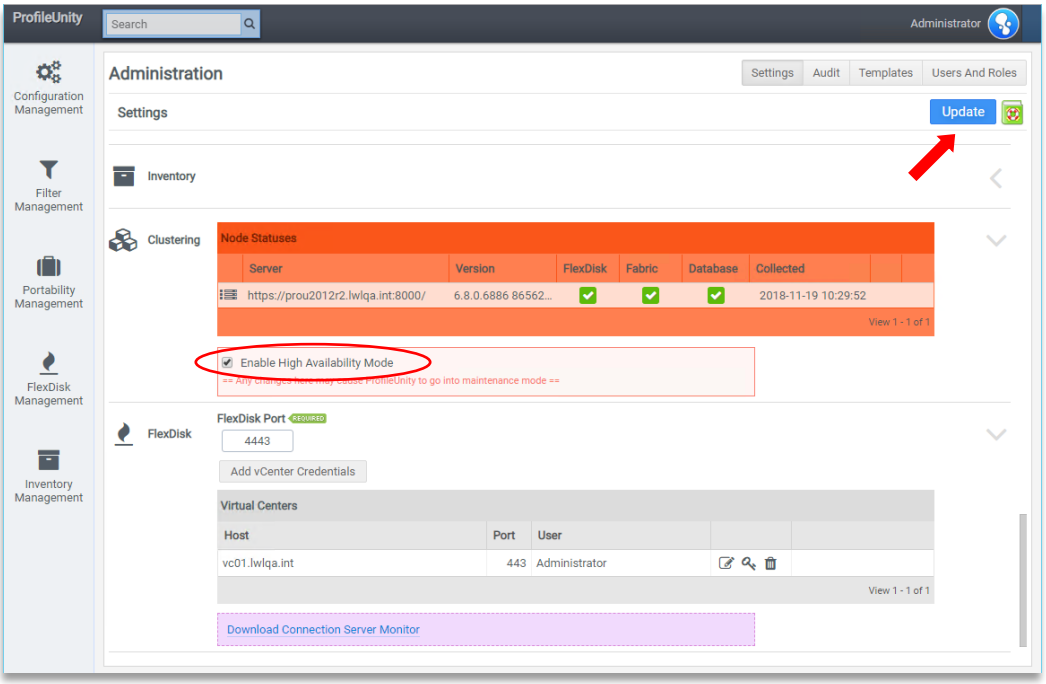
4. Install the Connection Server Monitor on all View Connection Brokers the users will be using to access pools. You will tell the Connection Server Monitor about this primary node installation of the ProfileUnity Management Console.
5. Go back to the FlexDisk section of ProfileUnity Management Console Administration Area to enter your VMware Virtual Center credentials. Click on the **Add vCenter Credentials** button and enter your credentials formatted the same as they are in the Virtual Center Client.





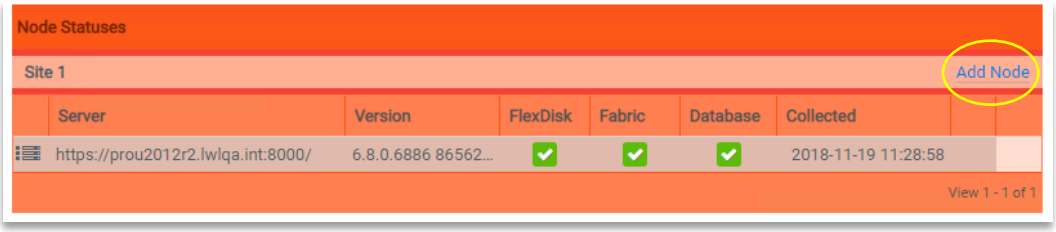
Adding additional nodes for ProfileUnity Clustering:

1. Install the ProfileUnity Management Console on the second server. This machine must:
    a. Be running Windows Server 2008 R2, 2012 R2 or 2016,
    b. Have a static IP address, and
    c. Be joined to a domain.
2. Go back to the first installation of the ProfileUnity Management Console on the primary server and go to the **Clustering** section of the Administration > Settings area as we are now creating the cluster of replicated nodes.
3. Check **Enable High Availability Mode** and then click on the blue **Update** button.

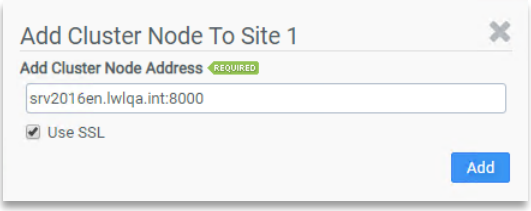


Please note that changing Enable High Availability Mode and updating the settings will put ProfileUnity temporarily into Maintenance Mode. When done, ProfileUnity will redirect you to the login screen where you can login and go back to the Administration > Settings area to complete the Clustering/FlexDisk configuration.

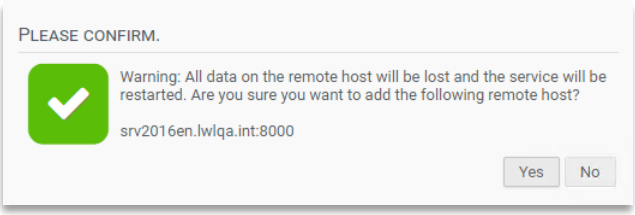
4. After logging back in to the Management Console on the primary server, go to **Administration > Settings > Clustering**. Click on the **Add Node** link which appears after updating the High Availability Mode setting.



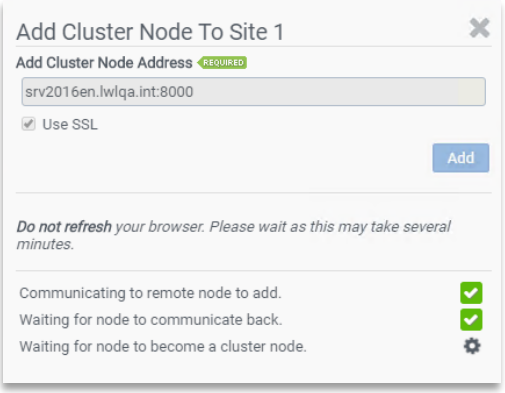
5. Enter the secondary node address and click on the **Add** button. This address should be the fully qualified domain name of the server where the second installation of the ProfileUnity Management Console was installed appended with the default port of 8000. For example, the node address would look like: prou2.mydomain.com:8000.



6. Confirm the addition of the second node to the service.



7. Please wait while the second node is added.



8. Install the ProfileUnity Management Console on the third server. This machine must:
    a. Be running Windows Server 2008 R2, 2012 R2 or 2016,
    b. Have a static IP address, and
    c. Be joined to a domain.
9. Go back to the first installation of the ProfileUnity Management Console on the primary server and go to **Clustering** section of the Administration > Settings area.
10. Click on the **Add Node** link.
11. Enter the node address for the third server in the cluster and click on the **Add** button. This address should be the fully qualified domain name of the server where the third installation of the ProfileUnity Management Console was installed appended with the default port of 8000. For example, the node address would look like: prou3.mydomain.com:8000.
12. Confirm the addition of the third node to the service.
13. Please wait while this node is added.
14. Repeat steps 8-13 to add additional nodes.

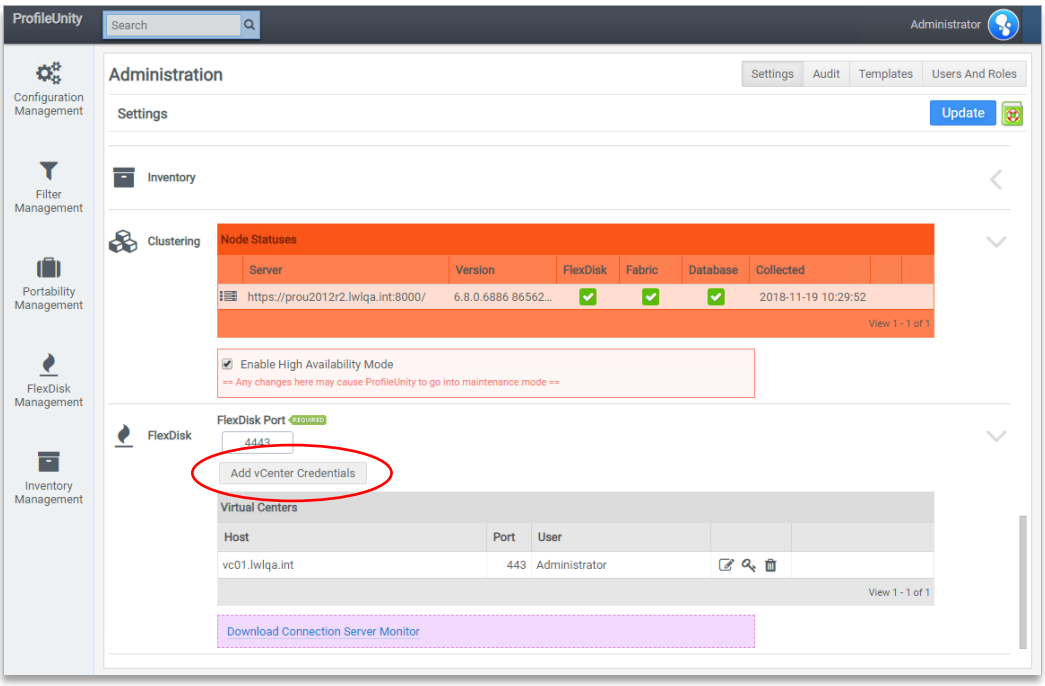
While setting up the FlexDisk service requires at least 3 nodes, you can certainly configure more nodes. However, please note that MongoDB requires that there be an ODD number of nodes. For example, if you already have 3 nodes configured and need to add another, please add 2 more for a total of 5, so that the total number of nodes at each site are an odd number.

Support for Multiple vCenter Servers

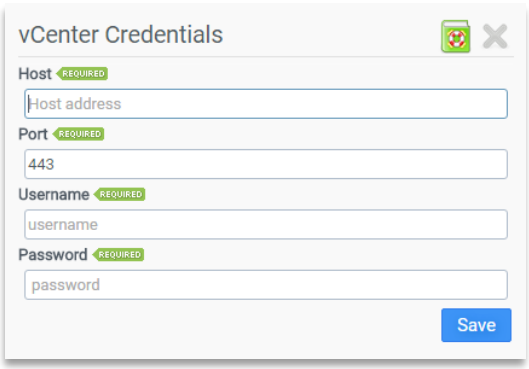
ProfileUnity's FlexDisk technology supports the usage of multiple VMware vCenter Servers. For organizations that have users that float between multiple vCenter Servers, FlexDisk coordinates communication and executes administrative tasks using the vCenter APIs to attach and detach VMDKs based on each user's requirements and the server to which they are currently attached.

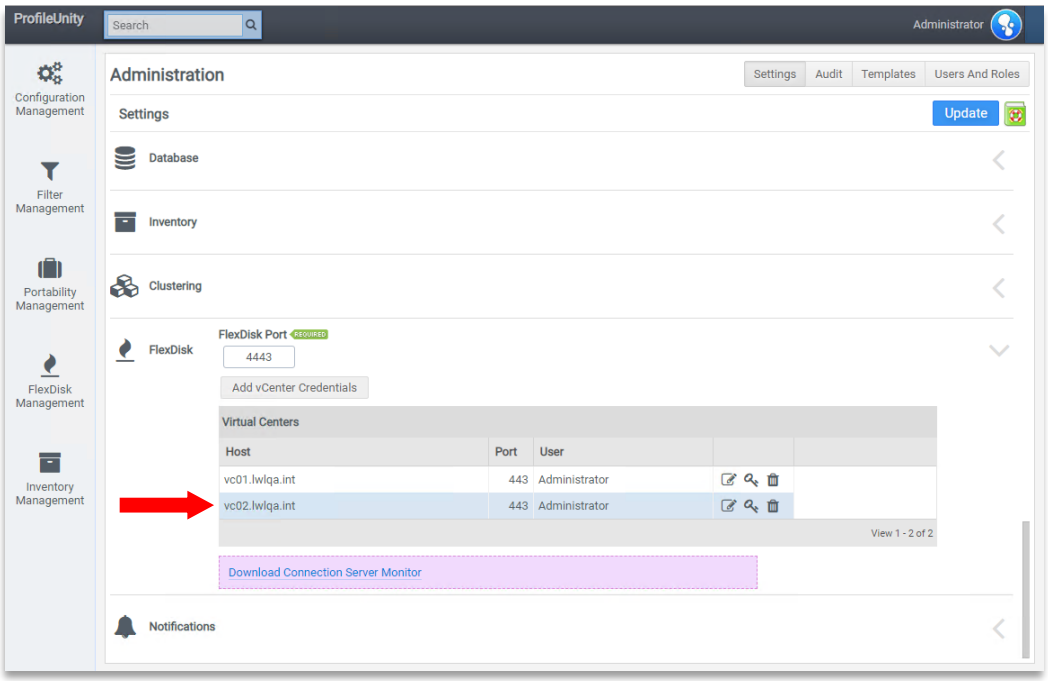
To configure for multiple VMware vCenter Servers:

1. Login to the ProfileUnity Management Console and go to the Administration area by going to your user name at the top right of the Management Console and selecting **Administration** from the drop-down list.
2. In the Settings tab, scroll down to the FlexDisk section and click on the **Add vCenter Credentials** button.



3. Enter your VMware Virtual Center credentials formatted the same as they are in the VMware vCenter Client.
4. Enter your VMware Virtual Center credentials formatted the same as they are in the VMware vCenter Client.



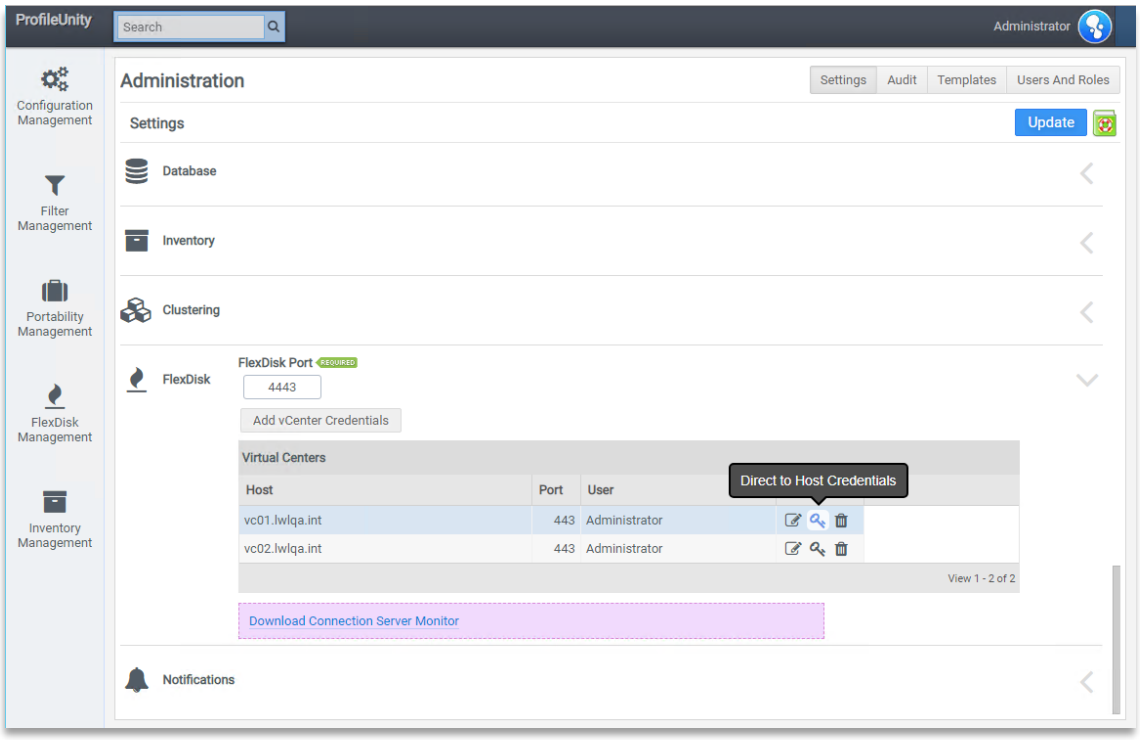

ProfileUnity
Search
Administrator
Configuration Management
Filter Management
Portability Management
FlexDisk Management
Inventory Management
Administration
Settings
Audit
Templates
Users And Roles
Settings
Update
Database
Inventory
Clustering
FlexDisk
FlexDisk Port
REQUIRED
4443
Add vCenter Credentials
Virtual Centers
Host
Port
User
vc01.lwlqa.int
443
Administrator
vc02.lwlqa.int
443
Administrator
View 1 - 2 of 2
Download Connection Server Monitor
Notifications

Direct to Host Mode

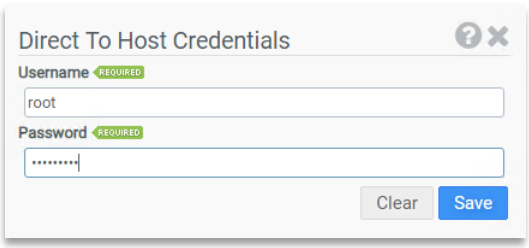
In the process of delivering VMDK-based layers, the FlexDisk technology makes calls to use the VMware vCenter APIs which is extra piece of software taking up additional resources. By using the Direct to Host Mode, FlexDisk can bypass vCenter and go directly to the host in order to speed up the attachment of VMDK layers and reduce overhead.

Direct to Host Mode can be configured in the FlexDisk Integration section of the Administration area after you enter your VMware vCenter Server credentials to add the server to the list of Virtual Centers for FlexDisk to use.

Click on the key icon to the right of your Virtual Center Host name.



Enter the root credentials for this vCenter Server, and click **Save**. Please note that all ESXi hosts under this vCenter Server have to have the same root user and password as the server in order for FlexDisk to have direct access to each host on this server.

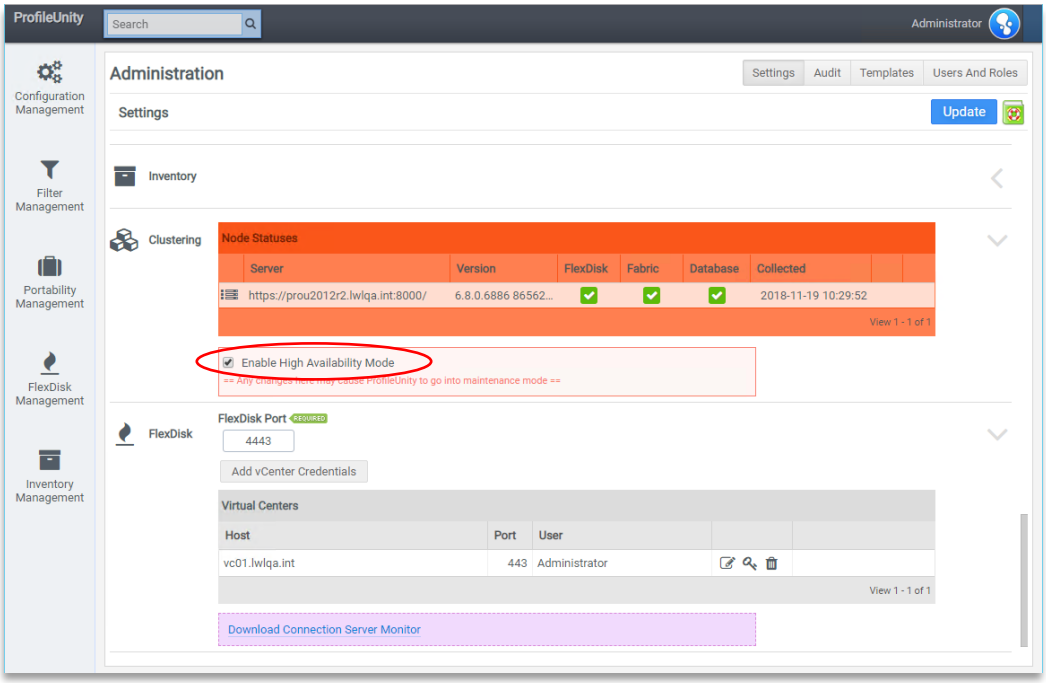


Using FlexDisk across Multiple Data Centers with One Master Console

The ProfileUnity FlexDisk architecture allows for multiple installations of ProfileUnity across multiple data centers. One master ProfileUnity Management Console will drive all the consoles for the other data centers. This applies to features like clustering and high availability. All traffic within a particular site will stay within that site while there is one master site that controls all the secondary sites.

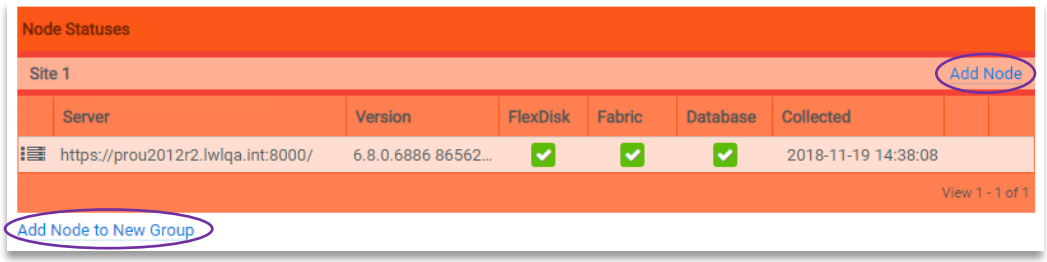
To configure nodes on multiple sites:

1. In the FlexDisk Integration section of the Administration Area, make sure that **Enable High Availability Mode** is checked. If it is not, check that option and click on the blue Update button.



Please note that changing Enable High Availability Mode and updating the settings will put ProfileUnity temporarily into Maintenance Mode. When done, ProfileUnity will redirect you to the login screen where you can login and go back to the FlexDisk Integration section in the Administration area to complete the FlexDisk configuration.

2. With **Enable High Availability Mode** on, notice the **Add Node** and **Add Node to New Group** links.



3. To add a new site, click on **Add Node to New Group** and enter the node address for another server located at another site and click on the **Add** button. This address should be the fully qualified domain name of the server where the ProfileUnity Management Console was previously installed appended with the default port of 8000. For example, the node address would look like: prou4.mydomain.com:8000.
4. To add a node to any site, click on the **Add Node** link within site's Node Status list and enter the node address for another server and click on the **Add** button. This address should be the fully qualified domain name of the server where the ProfileUnity Management Console was previously installed appended with the default port of 8000. For example, the node address would look like: prou4.mydomain.com:8000.

ProfileDisk: Full User Profile Delivery Employing Virtual Disks

Another powerful feature included in ProfileUnity is ProfileDisk. Very simply, ProfileDisk allows a user's entire Windows user profile to be contained on a virtual disk that can be attached and detached to a desktop as needed. The ProfileDisk stores 100% of all file system and registry changes in the user profile path so that all user settings and data are preserved on both persistent and non-persistent desktops. You won't have to spend time deciding which parts of the user's profile should be saved and made portable across your environment in an effort to reduce profile sizes. And since the user profile is available after mounting the drive at login rather than waiting for it to be streamed across the network, large profiles load much faster for your users.

As a user environment management solution, ProfileUnity has always delivered universally compatible user profiles across multiple Windows sessions. When a user logs on to a Windows session, his/her personal user profile settings are instantly pulled across the network in seconds. Customized settings such as application level customization, user-created spell checker data, Outlook signatures, desktop wallpaper, and much more are instantly made available for the user regardless if the machine is a VMware View, Citrix XenDesktop, thin client, or traditional Windows desktop.

However, the large amount of data stored in a user's profile tends to grow and not all of this data is necessary for a robust and complete user experience. As a result, ProfileUnity gives administrators granular control to set rules and choose what parts of the profile data to preserve and make portable with Portability Management and Settings. Reducing the user profile to contain only the essential settings and data speeds up logon times, reduces profile corruption instances, and eliminates the needless transfer of large amounts of data over the network.

Out-of-the-box, ProfileUnity comes with templates that cover 85% of the user profile and typical application and Windows settings. What about the rest of the settings? Administrators would have to create portability rules for any essential settings and applications that are not typically captured by ProfileUnity.

What if user login times are growing even while using Portability Management and Settings? Administrators are faced with either cutting more data from the profile to reduce streaming data over the network or telling their users to expect long logins.

In some cases, organizations want to provide users with their full user profile settings and data to give them a persistent desktop in a non-persistent environment. But again, user experience satisfaction decreases as the transfer of a large user profile causes delays at login.

ProfileUnity's ProfileDisk solves these issues. A ProfileDisk contains a user's entire profile on either a VHD or VMDK virtual disk. When the user logs in, that ProfileDisk is attached to their session. When the user logs out, the ProfileDisk is detached. This feature acts as an "easy button" or "catch all" for user profiles so that no settings are missed or forgotten. With the disk attached for the duration of the session, there is no need for the settings and data to be streamed across the network at login, resulting in the ability to support full persistence without compromising user experience.

Using a VMDK ProfileDisk with the new FlexDisk technology, the ProfileDisk can be attached to any View VDI desktop on the fly and detached at logoff on either a persistent or non-persistent desktop. This avoids the need to restore the profile at login and adds only about 5 seconds of login time for Virtual Center to hot-add the VMDK upon our request. This technology is also created in a way that preserves the native Windows format of the user profile.

Similarly, the same technique is used to attach a VHD ProfileDisk. The login process is held up for about 2 seconds to mount the VHD and then let Windows continue on its way. This technology supports Windows 7/10, Windows Server 2008 R2/2012 R2/2016 and the following desktop delivery types:

- VMware View VDI
- Citrix VDI, MCS and PVS
- Amazon Workspaces
- Physical desktops
- Multi-user operating systems like Microsoft RDS, Citrix XenApp and VMware View RDSH

While it might seem that employing ProfileDisk to capture the whole user profile would eliminate the need for or use of Portability rules, ProfileUnity's Portability Management can still be used in conjunction with ProfileDisk. Portability Management can help bridge the gap by capturing user profile settings from non-conforming applications that are not being saved in the standard local user folder or user's registry hive. Portability Management can also be used to help support multiple desktop sessions or multiple application sessions. When it comes to disaster recovery and using VMDKs for your users' profiles, Portability Management can be configured to save out the profile from the ProfileDisk VMDK to a CIFS share. Backing up and replicating a CIFS share may be easier then backing up and replicating a VMDK on a VMFS volume. Thus, Portability Management is still an administrator's powerful ally in customizing user profile management for their specific environment.

Configuring a VHD ProfileDisk

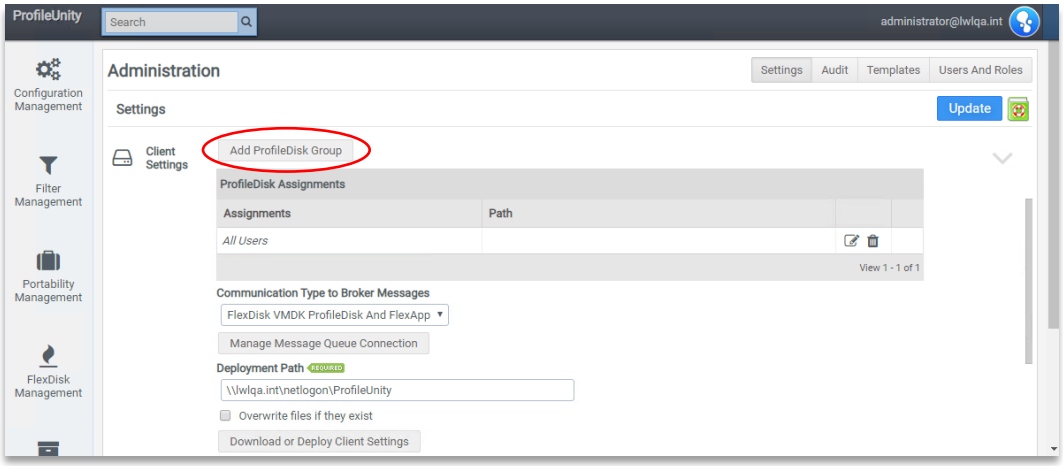
VHD-based ProfileDisks can be setup in two ways. The first method is to create a new Configuration using the Guided Configuration Wizard by selecting a "VHD ProfileDisk" template from the template library. One of the wizard's steps will allow you to configure VHD ProfileDisks similar to the instructions below. The second method is to configure VHD ProfileDisks in the Administration area using the instructions below.

User Group Management is available for VHD-based ProfileDisks. Different VHD-based ProfileDisks can be assigned to particular Active Directory User Groups. Also, each group of users can have separate paths to their own ProfileDisk so that loads can be balanced among different file shares.

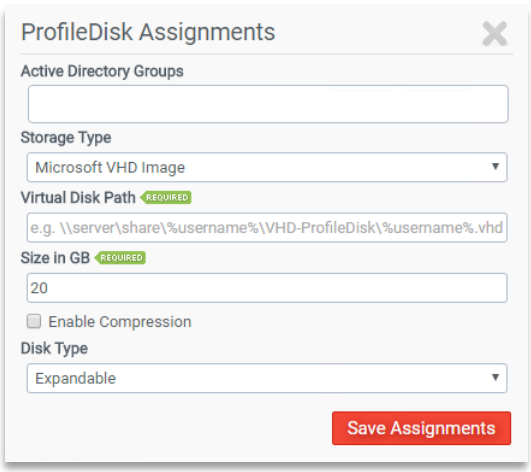
To use a VHD ProfileDisk to hold user profiles:

1. Login to the ProfileUnity Management Console and go to the Administration area. To get there, go to your user name at the top right of the Management Console and select **Administration** from the drop-down list. In the Settings tab, scroll down to the Client Settings section.

2. Add a new ProfileDisk and User Group assignment by clicking on the **Add ProfileDisk Group** button. Or you can edit an existing ProfileDisk assignment from the table below.



3. Configure your ProfileDisk settings and click **Save Assignments** when done.



   a. Enter the Active Directory Groups that will be assigned to this VHD-based ProfileDisk by typing the first few letters of the group and then selecting the correct group. Multiple groups may be selected. If this field is left blank, then the ProfileDisk will be assigned to all ProfileUnity users.
   b. Enter the **Virtual Disk Path** where the VHD will be located.
   c. Enter the VHD's maximum **Size in GB**.
   d. Select whether this disk is **Expandable** or **Fixed** under **Disk Type**.
4. Choose **VHD ProfileDisk** for the Fabric **Communication Type to Broker Messages**.
5. Enter the **Deployment Path** where the ProfileDisk configuration for all disks will be stored on your NETLOGON share.
6. Click the blue **Update** button in the upper right corner of the Management Console.
7. Click on the **Download or Deploy Client Settings** button to send a copy of the client configuration file to the specified location. Then you will be able to select your **Platform** for deployment. Choose "Domain" or "Cloud" to send the client configuration file to the specified **Deployment Path**. If not logged in as an Active Directory user, choose "Download" and manually place the client configuration file, clientsettings.xml, in the **Deployment Path**.
8. Scroll down to the ProfileUnity Tools section and click on **Download or Deploy Client Tools** to copy the ProfileUnity Client files to the specified location. Then you will be able to select your

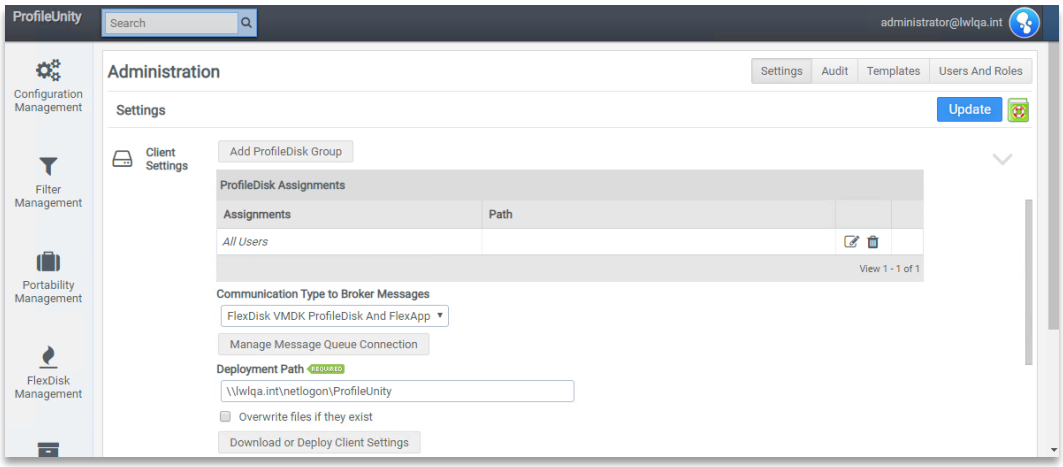
Platform for deployment. Choose “Domain” or “Cloud” to send the client configuration file to the specified **Deployment Path**. If not logged in as an Active Directory user, choose “Download” and manually place the Client Tools in the NETLOGON share on your domain controller.

9. Go to the Configuration Management area and create a configuration file. In order for ProfileDisk to work, you must have at least one ProfileUnity Configuration File (INI) downloaded or deployed to your **Deployment Path**. In order to configure ProfileDisk without other configuration settings, just create a blank configuration file in the Configuration Wizard by choosing a “Manual” configuration with no other settings in it.
10. Reboot all user desktops. If Microsoft® .NET Framework 4.6.2 is not installed in the base operating system on a particular desktop, it will be installed once the desktop restarts. The installation of .NET 4.6.2 will increase the time it takes before the desktop is available again.

Configuring a VMDK ProfileDisk

To use a VMDK ProfileDisk to hold user profiles:

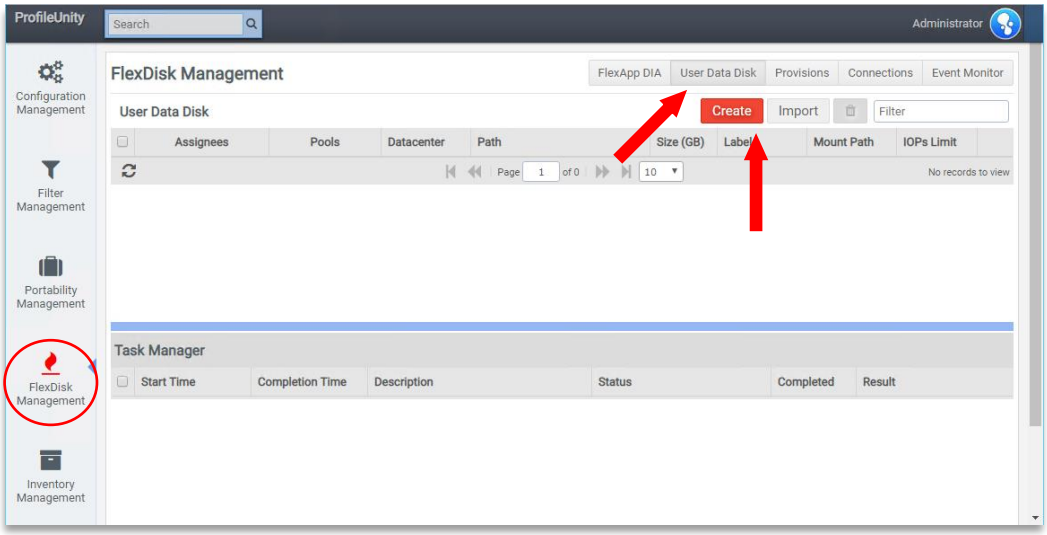
1. When installing the ProfileUnity Management Console, you were asked to pick a broker for the integration of the FlexDisk VMDK distribution technology. FlexDisk will govern the operation of the VMDK ProfileDisk whether you use **Citrix**, **VMware**, **Microsoft WVD** or **Other** as your broker.
2. Login to the ProfileUnity Management Console and go to the Administration area. To get there, go to your user name at the top right of the Management Console and select **Administration** from the drop-down list.
3. If you selected VMware as your broker, you will need to setup the FlexDisk service if you haven’t already done so. Please follow the directions given in the **How to Setup ProfileUnity Clustering for FlexDisk** section earlier.
4. In Administration > Settings, scroll to the Client Settings section and:

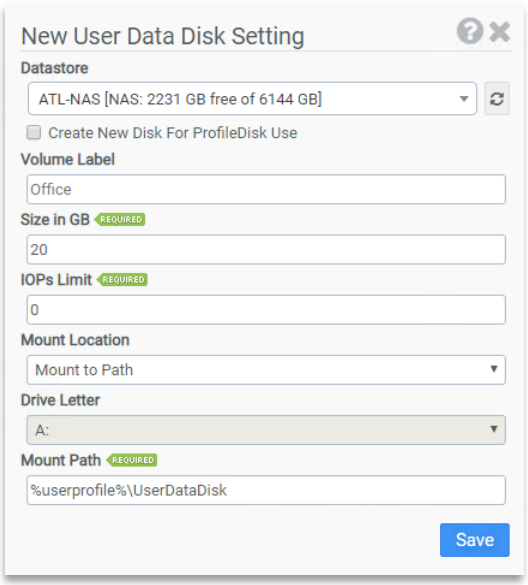


   a. Set **Communication Type to Broker Messages** to either “FlexDisk VMDK ProfileDisk Only” or “FlexDisk VMDK ProfileDisk And FlexApp” depending on which technologies you will be using.
   b. Set the **Deployment Path** to your NETLOGON path.
   c. Click the blue **Update** button in the upper right corner of the Management Console.
   d. Click on the **Download or Deploy Client Settings** button to send a copy of the client configuration file to the specified location. Then you will be able to select your **Platform** for deployment. Choose “Domain” or “Cloud” to send the client configuration file to the specified **Deployment Path**. If not logged in as an Active Directory user, choose

"Download" and manually place the client configuration file, clientsettings.xml, in the **Deployment Path**.

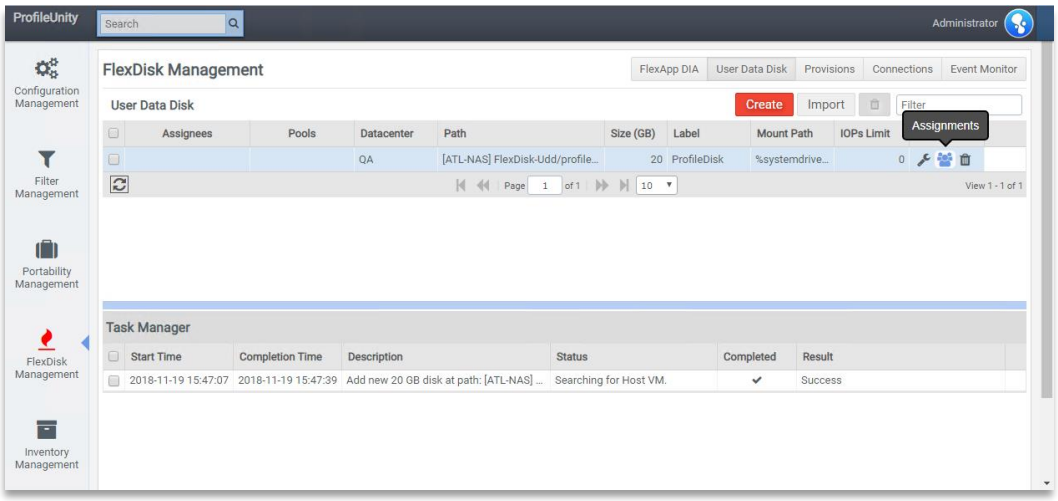
5. Scroll down to the ProfileUnity Tools section and click on **Download or Deploy Client Tools** to copy the ProfileUnity Client files to the specified location. Then you will be able to select your **Platform** for deployment. Choose "Domain" or "Cloud" to send the client configuration file to the specified **Deployment Path**. If not logged in as an Active Directory user, choose "Download" and manually place the Client Tools in the NETLOGON share on your domain controller.
6. Go to the Configuration Management area and create a configuration file. In order for ProfileDisk to work, you must have at least one ProfileUnity Configuration File (INI) downloaded or deployed to your **Deployment Path**. In order to configure ProfileDisk without other configuration settings, just create a blank configuration file in the Configuration Wizard by choosing a "Manual" configuration with no other settings in it.
7. Reboot or refresh all user desktops. If Microsoft® .NET Framework 4.6.2 is not installed in the base operating system on a particular desktop, ProfileUnity will install it once the desktop restarts. The installation of .NET 4.6.2 will increase the time it takes before the desktop is available again.
8. In the ProfileUnity Management Console, click on **FlexDisk Management** on the left navigation panel. Click on the **User Data Disk** tab and then click on the red **Create** button to provision a user data disk:



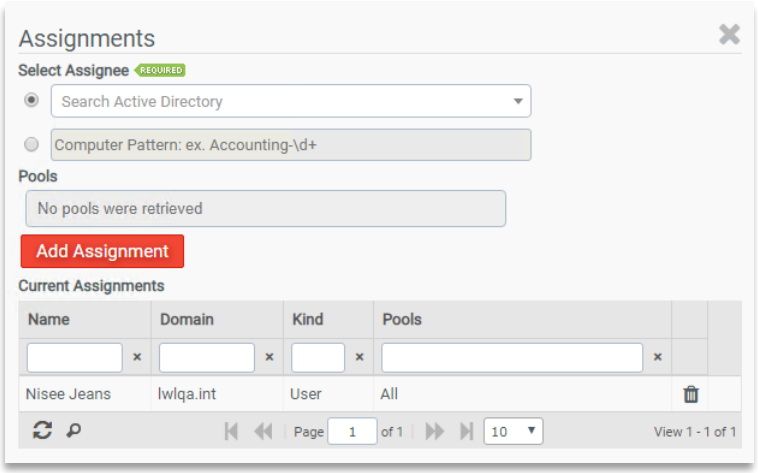


a. Pick your **Datastore**.
b. Check **Create New Disk for ProfileDisk Use**.
c. Change the **Size in GB** if needed.
d. Click **Save**.
e. After the Task Manager list show the disk creation is successful, refresh the User Data Disk list if necessary.

9. With the user data disk just created, click on the Assignments icon to create an assignment for users or groups.



10. Select your user or group. Pools will automatically populate if you have VMware View and have installed the Connection Service Monitor. Click on **Add Assignment** to add it to the list of current assignments. When done, click on the 'X'.



11. With the VMDK ProfileDisk configuration complete, the user(s) should be able to login.

Application Layering with ProfileUnity's FlexApp Technology

ProfileUnity's FlexApp technology provides an alternative way of delivering applications to just the users who need them. FlexApp allows Administrators to assign department-level applications to groups of users, and to, optionally, empower selected users to install their own applications. By making use of virtual disks (VHDs or VMDKs) 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.

As an alternative application delivery method, FlexApp compliments other solutions and can be used in addition to VMware ThinApp, Microsoft App-V, and Citrix XenApp as needed to meet specific application requirements. Unlike some of the other application virtualization solutions, FlexApp uses the technique of application redirection versus application encapsulation. Encapsulated applications are isolated from the operating system with the intention that they can function in a self-contained environment. This isolation can often lead to reduced application incompatibilities as well as the potential launch capability across multiple operating systems. However, once an application has been isolated from the host operating system, communication issues between applications can become more prevalent. The FlexApp Layering technology, leverages symbolic links to the application binary files inside the OS where that application would normally be installed, so that applications can be inserted into the user's mix as needed. FlexApp applications are not isolated and thus look native to the OS and any other dependent applications.

In addition, FlexApp can be combined with ProfileUnity's Privilege Elevation & Application Restrictions modules to provide Application Rights Management (ARM) that enables administrators to securely grant specific users detailed application rights without making them a Windows Administrator.

The release of ProfileUnity 6.5 ushered in FlexApp 2.0. ProfileUnity's initial FlexApp Technology design allowed for user installed applications (UIA) to snap-in as part of the user's profile and administrators to deliver department installed applications (DIA) as additional applications apart from the base image to departments or segments of users without maintaining multiple desktop images for different sets of users. But this was all for non-persistent virtual desktops. FlexApp 2.0 technology has been re-architected for use on both persistent and non-persistent desktops. In making FlexApp 2.0 a truly non-persistent application layer based on a virtual registry and a virtual file system overlay that is not overwriting the real registry and file system settings. FlexApp 2.0 can be used on physical, RDSH, persistent, and non-persistent machines. The re-architecture of FlexApp 2.0 has resulted in faster load times for FlexApp applications and a greater compatibility with even more applications.

FlexApp Department Installed Applications (DIA)

Department installed applications include those that are still managed by IT staff or desktop administrators and delivered to users but are not included in a master image. Before delivering these applications to specific users or departments, administrators package applications using the FlexApp Packaging Console. When packaging an application using the FlexApp Packaging Console, each application gets its own virtual disk. Once the packages have been created, the admin assigns the packages to users from within the FlexApp DIA Configuration Module in the ProfileUnity Management Console. When a user logs in to their desktop, ProfileUnity, using its filter system, breaks down which applications have been assigned to the user by the administrator. These applications are then played back into the OS and appear alongside of their golden image applications.

FlexApp User Installed Applications (UIA)

During a user's session, FlexApp is designed to load in front of any application installation that is run by the user. When an application installation is detected, FlexApp intercepts the file and redirects it to the target virtual disk during the installation process. FlexApp then records this event for future playback. All file and registry calls are also intercepted and recorded on the network virtual disk. Each user has his or her own VHD or VMDK in which to store user installed applications. FlexApp exchanges application folders with shortcuts so that an application is able to run from a virtual disk that resides on the network rather than the virtual machine.

Essentially, the application installation process is virtualized during the install capture. Because of this sequence, once the application is installed or played back, FlexApp does not need to run any software in between the OS and the application. FlexApp also keeps the applications running native to the OS, which is a significant advantage as our application compatibility is extremely high with this technology.

> ***Note: Running UIA at the same time as DIA is no longer supported. As a preferred technology, the FlexApp DIA configuration module is displayed by default while the FlexApp UIA configuration module is hidden. To use the FlexApp UIA configuration module, go to Administration > Settings and uncheck the "Hide FlexApp UIA Module" option in the Miscellaneous section.***

Click-To-Layer

ProfileUnity's FlexApp Technology layers in application packages in parallel to the user login process without impacting user login speeds. It generally takes 2-4 seconds to layer in each package and make it ready for use. In cases where 10 or more FlexApp packages are assigned to a user, it is possible for the login process to finish and the desktop to be ready before all the FlexApp packages have been layered in leaving a user waiting for their applications to appear. ProfileUnity's Click-To-Layer feature allows admins to delay selected FlexApp packages and layer them in on an as-needed basis only when a user clicks on the icon to open that application. Heavily used or large applications can still be configured to be layered in during the user login process. ProfileUnity's default action is to layer in applications in parallel to the user login process.

Combining FlexDisk with FlexApp

FlexDisk can be used in conjunction with FlexApp to achieve better performance with application layering. Segmenting applications into different layers allows administrators to update application layers individually and push out layers to selective groups of users. Using this technique, each application layer is stored in its own VMDK and each user who is assigned to this application layer is given their individual write cache using a snap shot on the VMFS storage volume. Since the application is running from a VMDK on the same or similar storage as the VDI session, application performance will be very similar to an application baked into the base OS.

Combining FlexDisk with FlexApp results in a performance boost and speedier logons. FlexDisk applications are self-contained within each VMDK and connected to the user's desktop when provisioned by your desktop broker. FlexDisk can connect up to 58-60 applications to a user if each application has its own

VMDK. Otherwise you can combine applications into a single VMDK as well. The same FlexDisk application can be shared among hundreds of users.

Folder Redirection with FlexApp

Using the Guided Configuration Wizard within ProfileUnity will allow you to automatically configure all the settings appropriately for FlexApp. The Guided Configuration Wizard options will virtualize the majority of the user's profile and provide the optimum configuration for portability and folder redirection.

FlexApp DIA and UIA System Requirements

- Windows® 7 or 10 — 32-bit or 64-bit
- Capturing Microsoft® .NET Frameworks are not supported by FlexApp, so you will need to put the needed frameworks in your base image. Since you will probably not be able to predict what applications your users are installing, placing 3.5 SP1 and 4.6.2 .NET Framework in the base image is the best strategy.
- Your capture path can be a network VHD which is managed by ProfileUnity or a VMDK.
- Windows must be activated and genuine.
- FlexApp DIA supports Windows Server 2008 R2, 2012 R2, 2016, and 2019 in Remote Desktop Service mode, RDSH, and TSE. In addition to supporting non-persistent desktops, FlexApp DIA also supports well-managed persistent desktops.
- FlexApp UIA only supports non-persistent virtual desktops running Windows 7.
- Running both FlexApp DIA and FlexApp UIA at the same time is not supported.

FlexApp DIA and UIA Supported Applications and Non-Supported Applications

Supported Application Types

- ActiveX
- Browser Plug-Ins
- Office Plug-Ins
- COM+
- COM DLL
- Services
- Apps That Use AppData Local
- Apps That Use AppData Roaming
- Shell Extensions
- Shell Content Menus
- Software Based Drivers
    - PDF Printers
    - Image Printers
- Applications That Need Updated Fonts In The System
- MSI Based Installers
- EXE's Based Installers
- EXE's That Spawn MSI's
- Screen Savers
- Shell Replacements
- Antivirus
    - Antivirus programs work, if drivers work with "On Service Start" at login.

Non-Supported Application Types

- Applications That Edit The Local Group Policy
- Drivers Needed On Boot
- Antivirus
    - If the antivirus software needs its driver on boot, it is currently not supported, but most work with FlexApp "On Service Start" for its driver at login.
- Applications where the licensing is tied to the computer it was installed on.

Using FlexApp DIA with Cloud Storage, VHDs or Local Disks

Department installed applications include those that are still managed by IT staff or desktop administrators and delivered to users but are not included in a master image. ProfileUnity with FlexApp DIA is enabled by the administrator for select users or groups within the ProfileUnity Management Console. Admins package applications which are provisioned a virtual disk to run separately from the OS and local session. When users login, these FlexApp applications appear alongside of their golden image applications.

> ***One of the most powerful features of ProfileUnity is the filters that can be leveraged during the assignment of FlexApp Layers. Please note that filters apply only to cloud, VHD-based or local disk FlexApp packages/layers. VMDK-based FlexApp layers currently support user, group and basic machine assignments. The primary reason for the difference in VHD vs. VDMK package types centers on architecture. The ProfileUnity Configuration INI-based workflow does not apply to the VMDK-based assignments controlled by vCenter.***

Installing the FlexApp Packaging Console & Packaging Applications

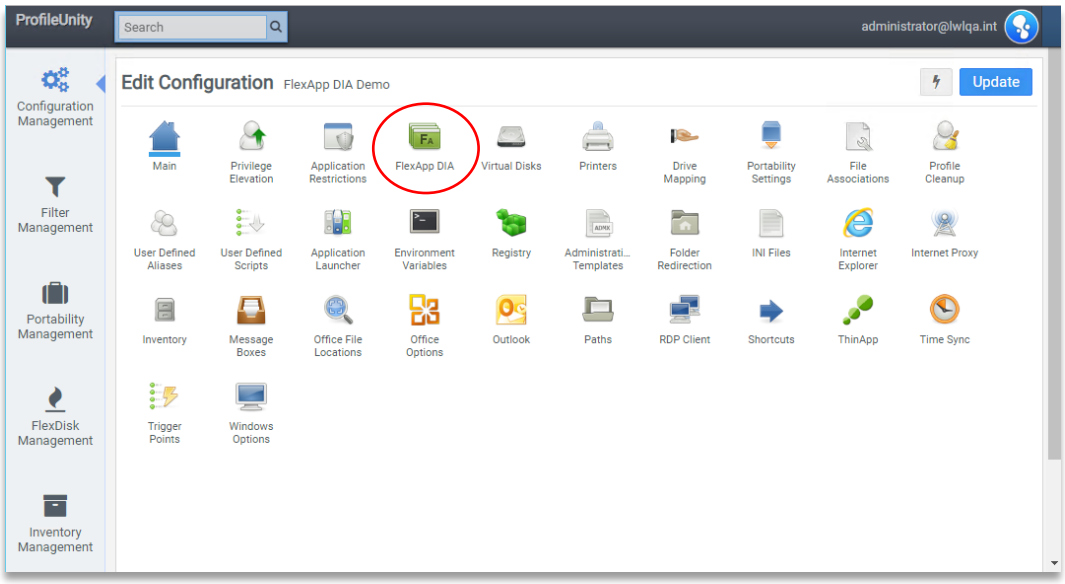
The FlexApp Packaging Console allows administrators to configure and prepare any applications that will need to be configured for users and made available as a department installed application (DIA). This software can be downloaded from the ProfileUnity Management Console and installed on a separate machine for packing applications. For more information on installing and using the FlexApp Packaging Console, please see the ***FlexApp Packaging Console Manual***.

Enabling Cloud, VHD-based, or Local Disk FlexApp DIA Layers for Users in the Management Console

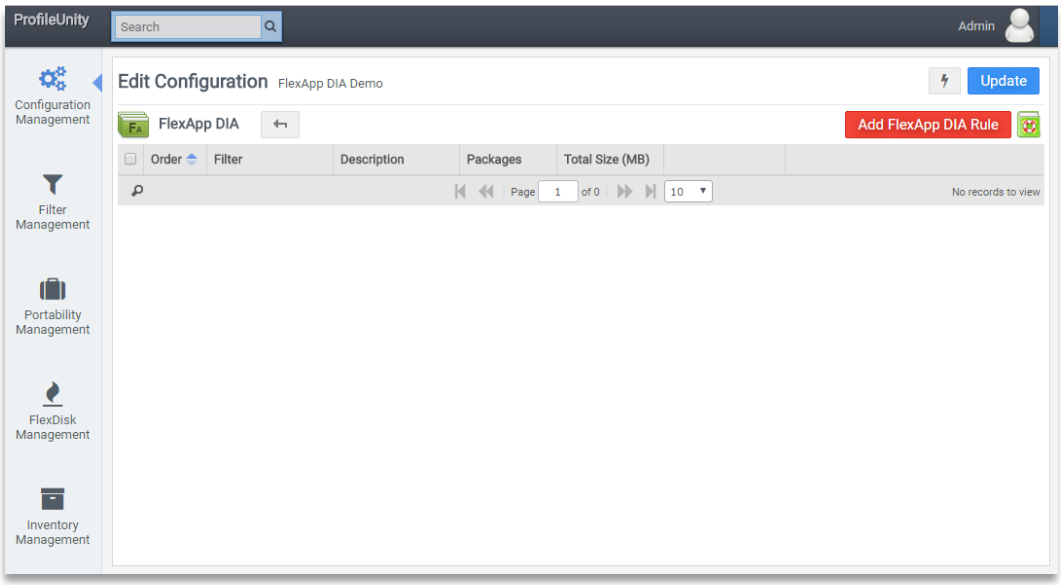
After applications have been packaged on the appropriate 32-bit or 64-bit OS platform using the FlexApp Packaging Console, the ProfileUnity Management Console enables admins to set which applications are configured for which users or user groups.

1. Select **Configuration Management** from the navigation tree on the left-hand side of the ProfileUnity Management Console.
2. Create a new configuration or choose the configuration you wish to edit and click on the **Edit** icon to the right of the configuration name.

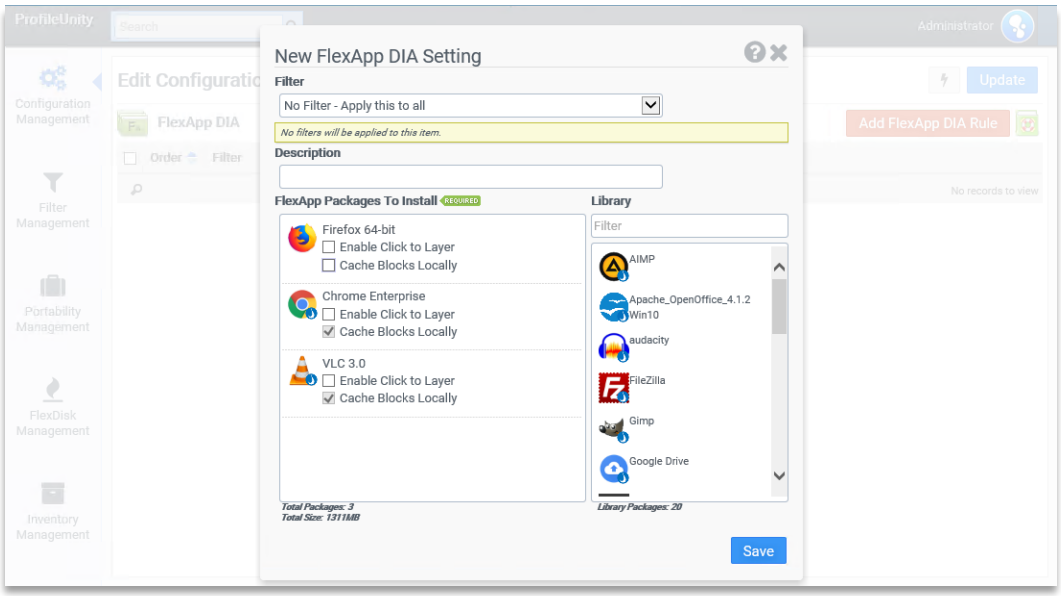
3. To setup cloud, VHD-based or local disk FlexApp DIA, click on the **FlexApp DIA** icon.



4. Click on **Add a FlexApp DIA Rule** button.



5. Choose your FlexApp packages for this DIA Rule.



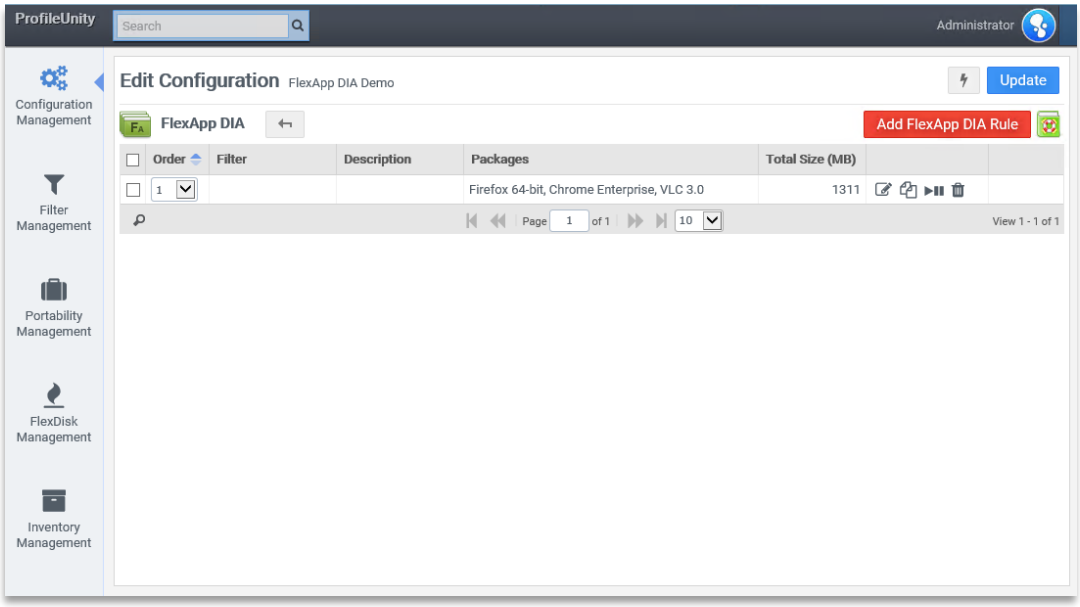
   a. Optionally select a **Filter** for this configuration element to apply this FlexApp DIA rule to certain users or groups.
   b. Then drag over the applications listed in the library that you want FlexApp to add in when the user logs into his session.
   c. Check **Enable Click To Layer** if you wish to delay the layering of this application from user login to upon user request after clicking on the application icon on the desktop. For large applications that can take a longer amount of time before they are ready for use, we recommend that you do NOT select **Enable Click To Layer** and instead layer those applications during the user login process to decrease any user wait time.

> ***Note: If the desired application has been packaged prior to FlexApp Packaging Console v6.7, the package can be automatically updated for Click-to-Layer compatibility by opening the FlexApp Packaging Console version 6.7 or higher, selecting 'Edit the package's metadata' for the desired package and clicking OK.***

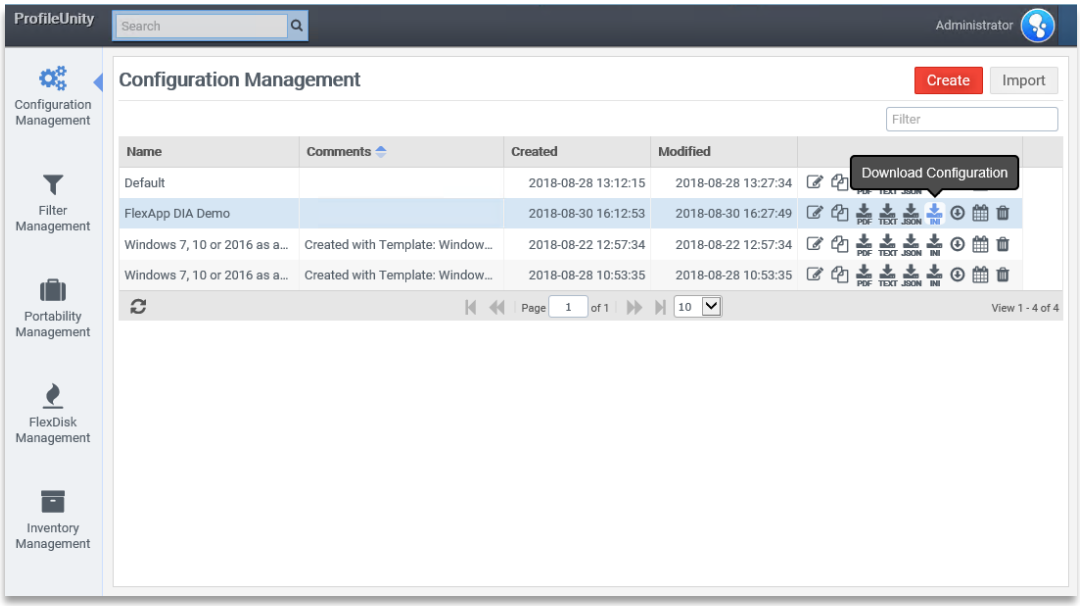
   d. Check **Cache Blocks Locally** if desired. FlexApp Packages can be configured to replay using a cached mode on user desktops which is particularly helpful with inconsistent network connections, SMB shares behind firewalls, and routers that have many hops. In cached mode, ProfileUnity will request blocks of the FlexApp package and store them locally so the next time the block is referenced, it is read from local storage. **Cache Blocks Locally** is enabled by default for FlexApps on cloud storage and is optional for FlexApps on SMB storage.

6. Click **Save**.

7. Click on the blue **Update** button in the upper right-hand corner to update the configuration with the FlexApp DIA changes.

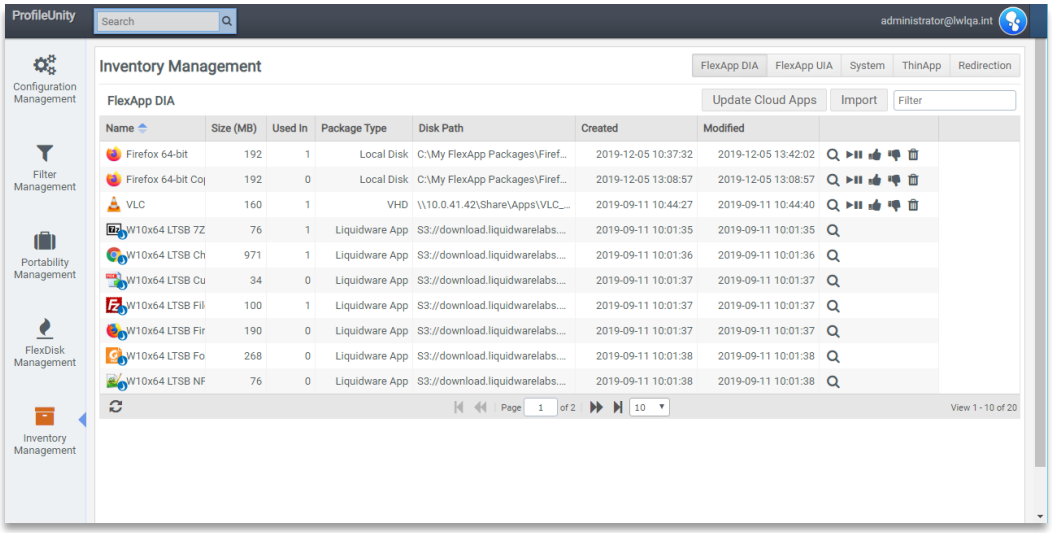


8. To activate your changes for any subsequent user logins, remember to download a new INI file after you have made all changes to your configuration.



Viewing Cloud, VHD-based or Local Disk FlexApp DIA Packages in the Management Console

1. Select **Inventory Management** from the navigation tree on the left-hand side of the ProfileUnity Management Console.
2. Make sure **FlexApp DIA** is selected at the top of the main window.
3. All the applications that have been packaged and created as cloud, VHD or local disk types to be used by FlexApp will be listed in the main window.



Using FlexApp DIA with VMDKs

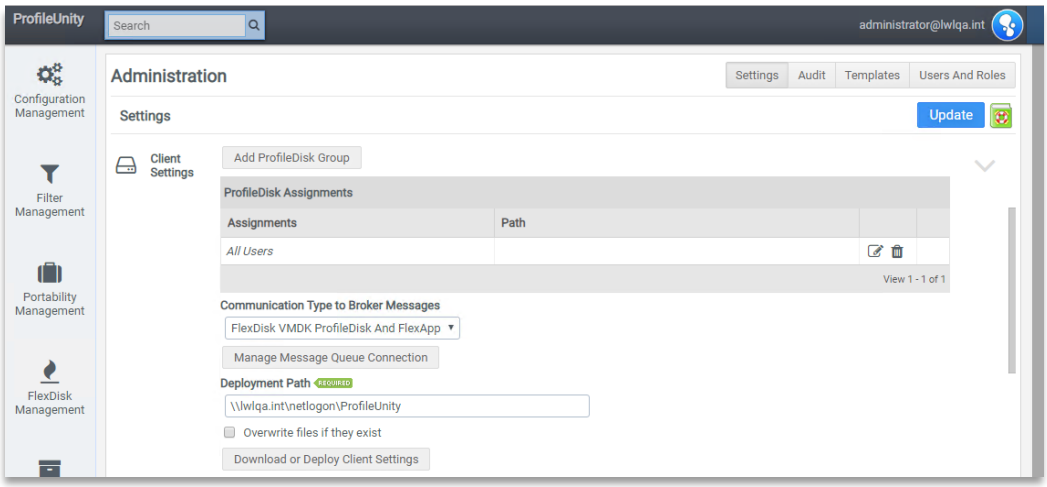
Department installed applications include those that are still managed by IT staff or desktop administrators and delivered to users but are not included in a master image. ProfileUnity with FlexApp DIA is enabled by the administrator for select users or groups within the ProfileUnity Management Console. Admins package applications which are provisioned a virtual disk to run separately from the OS and local session. When users login, these FlexApp applications appear alongside of their golden image applications.

> ***One of the most powerful features of ProfileUnity is the filters that can be leveraged during the assignment of FlexApp Layers. Please note that filters apply only to cloud, VHD-based or local disk FlexApp packages/layers. VMDK-based FlexApp layers currently support user, group and basic machine assignments. The primary reason for the difference in VHD vs. VDMK package types centers on architecture. The ProfileUnity Configuration INI-based workflow does not apply to the VMDK-based assignments controlled by vCenter.***

Starting the FlexDisk Service for VMDK-based Application Delivery

To use VMDKs for application delivery:

1. When installing the ProfileUnity Management Console, you were asked to pick a broker for the integration of the FlexDisk VMDK distribution technology. FlexDisk will govern the operation of the VMDK ProfileDisk whether you use **Citrix**, **VMware**, **Microsoft WVD** or **Other** as your broker.
2. Login to the ProfileUnity Management Console and go to the Administration area. To get there, go to your user name at the top right of the Management Console and select **Administration** from the drop-down list.
3. If you selected VMware as your broker, you will need to setup the FlexDisk service if you haven't already done so. Please follow the directions given in the **How to Setup ProfileUnity Clustering for FlexDisk** section earlier.
4. In the Settings tab, scroll to the Client Settings section in the Administration area and:



   a. Set **Communication Type to Broker Messages** to "FlexDisk VMDK ProfileDisk and FlexApp".
   b. Set the **Deployment Path** to your NETLOGON path.

c. Click the blue **Update** button in the upper right corner of the Management Console.
d. Click on the **Download or Deploy Client Settings** button to send a copy of the client configuration file to the specified location. Then you will be able to select your **Platform** for deployment. Choose "Domain" or "Cloud" to send the client configuration file to the specified **Deployment Path**. If not logged in as an Active Directory user, choose "Download" and manually place the client configuration file, clientsettings.xml, in the **Deployment Path**.

5. Scroll down to the ProfileUnity Tools section and click on **Download or Deploy Client Tools** to copy the ProfileUnity Client files to the NETLOGON share on your domain controller.
6. Go to the Configuration Management area and create a configuration file. In order for the FlexDisk Service to work, you must have at least one ProfileUnity Configuration File (INI) downloaded to your **Deployment Path**. In order to configure FlexDisk without other configuration settings, just create a blank configuration file in the Configuration Wizard by choosing a "Manual" configuration with no other settings in it.
7. Reboot or refresh all user desktops. If Microsoft® .NET Framework 4.6.2 is not installed in the base operating system on a particular desktop, ProfileUnity will install it once the desktop restarts. The installation of .NET 4.6.2 will increase the time it takes before the desktop is available again.
8. Now you are ready to install the FlexApp Packaging Console in a separate location.

Installing the FlexApp Packaging Console & Packaging Applications

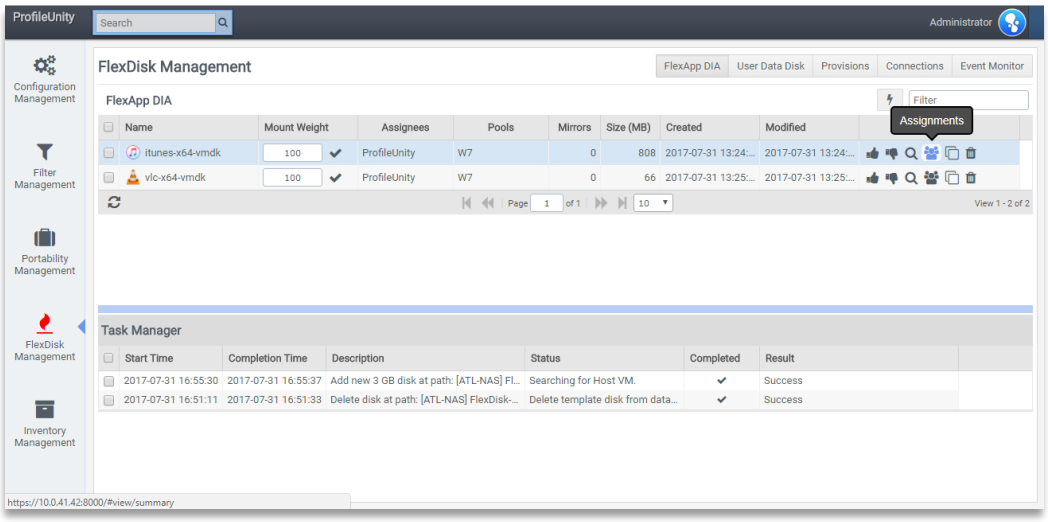
The FlexApp Packaging Console allows administrators to configure and prepare any applications that will need to be configured for users and made available as a department installed application (DIA). This software can be downloaded from the ProfileUnity Management Console and installed on a separate machine for packing applications. For more information on installing and using the FlexApp Packaging Console, please see the ***FlexApp Packaging Console Manual***.

When using the FlexApp Packaging Console to create VMDK-based application packages, select "FlexDisk" as the Package Type and choose which FlexDisk datastore to which the package will be saved.

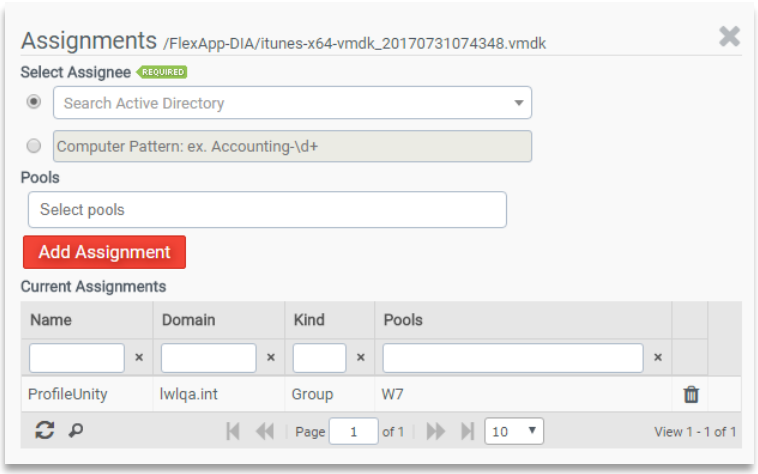
Assigning VMDK-based FlexApp DIA Layers to Users, Groups or Computers

After applications have been packaged on the appropriate 32-bit or 64-bit OS platform using the FlexApp Packaging Console, the ProfileUnity Management Console enables admins to set which applications are assigned to which users, user groups or computers. Computer-level assignments allow for pattern-based machine name matching to cover a pool of machines rather than a single machine. VMDK-based FlexApp packages can be used with or without ProfileUnity as a Service. If ProfileUnity as a Service is used, FlexApp packages are played back on service boot making them available before the user logs in. If ProfileUnity as a Service is not used, playback of the FlexApp packages occurs at user login. However, no user assignments are needed in this case which may be valuable in a lab scenario where many users have access to login to a finite pool of computers.

1. Select **FlexDisk Management** from the navigation tree on the left-hand side of the ProfileUnity Management Console, and select the FlexApp DIA tab at the top.
2. Click on **Assignments** for each FlexApp VMDK that needs to be assigned to users.

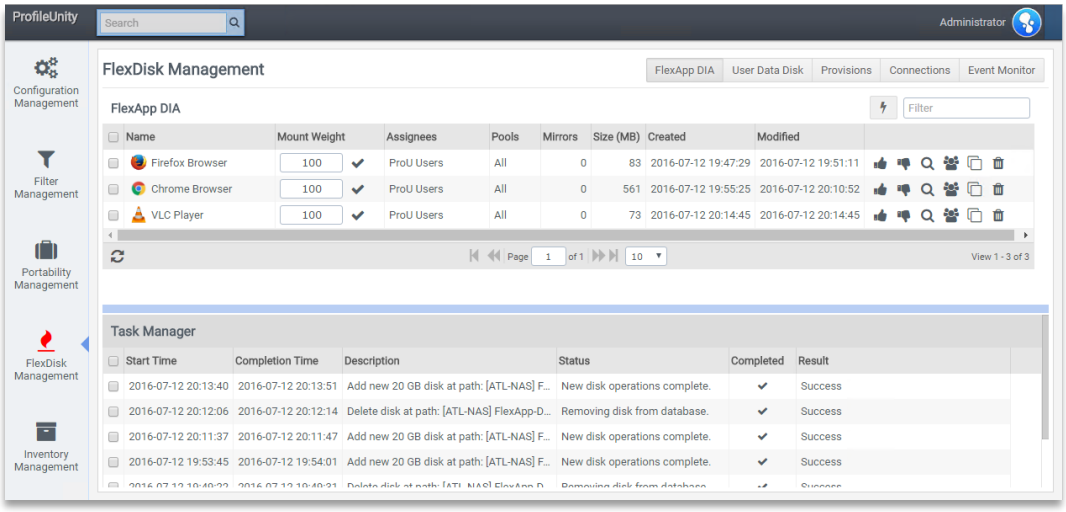


3. Either search Active Directory to select a user or group from the list for user assignments, or enter a machine name pattern for computer-level assignments. If you have VMware View and have installed the Connection Service Monitor, you will get the option to limit the application to the user, group, computer and/or the pool. Click on **Add Assignment** to add it to the list of current assignments. When done click on the 'X'.



Viewing VMDK-based FlexApp DIA Packages in the Management Console

1. To see your FlexDisk packages, select **FlexDisk Management** from the navigation tree on the left-hand side of the ProfileUnity Management Console and make sure **FlexApp DIA** is selected at the top of the main window.

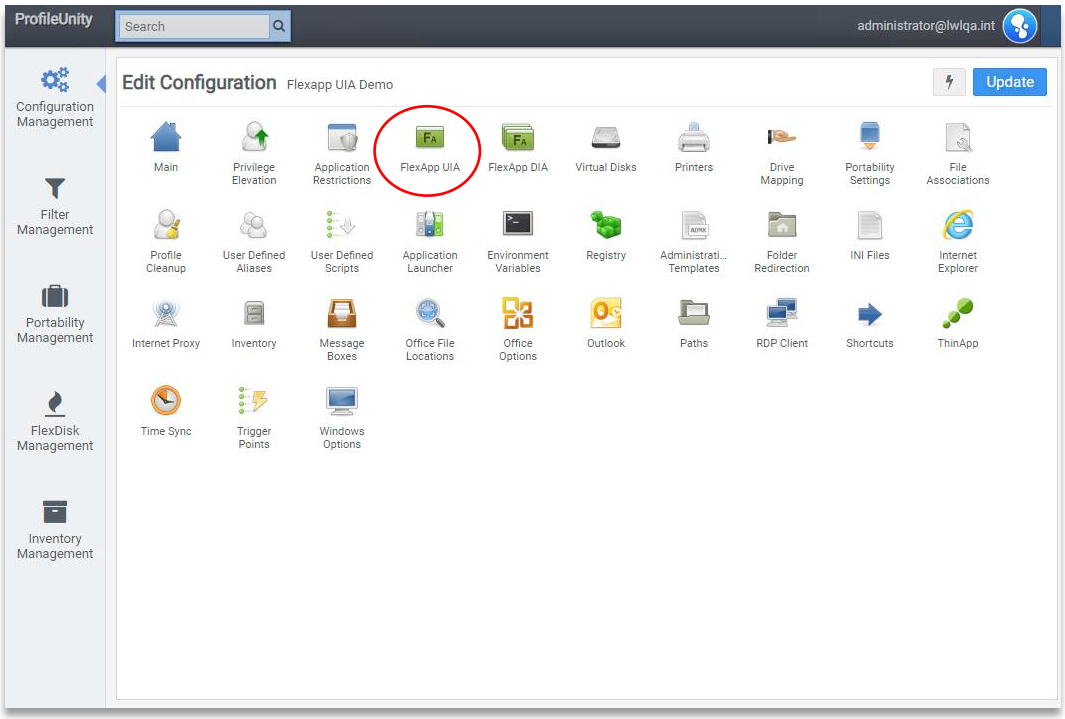


Using FlexApp UIA with VHDs or Persistent Disks

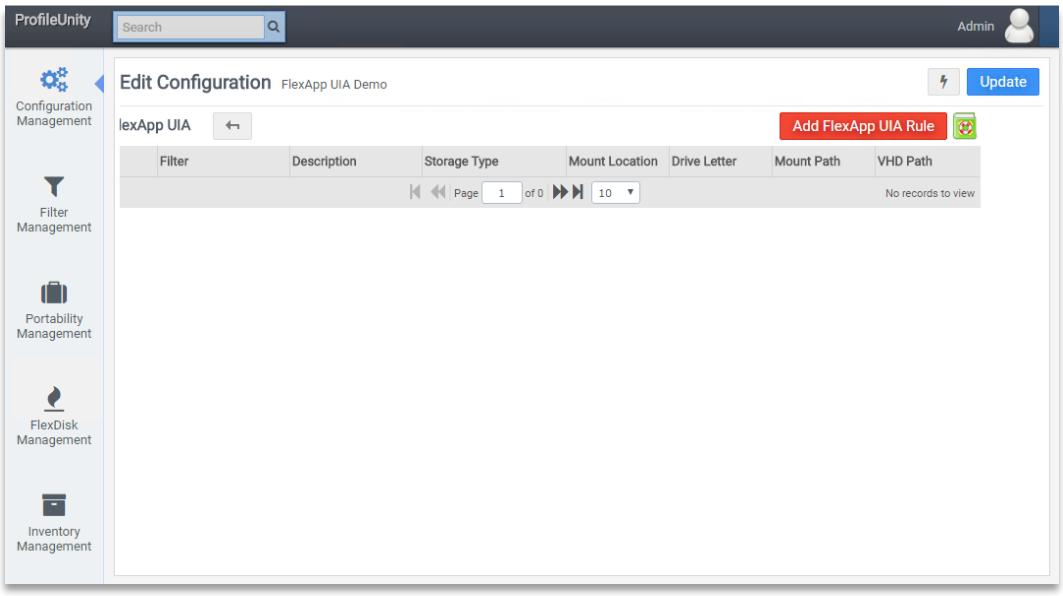
User installed applications are those applications that are installed and managed by the individual user rather than supported by the IT staff. ProfileUnity with FlexApp UIA is enabled by the administrator for select users or groups within the ProfileUnity Management Console. Users are provisioned a virtual disk to store their applications separately from the OS and local session. Users install applications just as they normally would. During this process, ProfileUnity with FlexApp UIA "hooks" the installation of the program, adding specialized "links" both into the local Windows OS and into the user's managed ProfileUnity with FlexApp settings.

Enabling VHD-based FlexApp UIA for Users in the Management Console

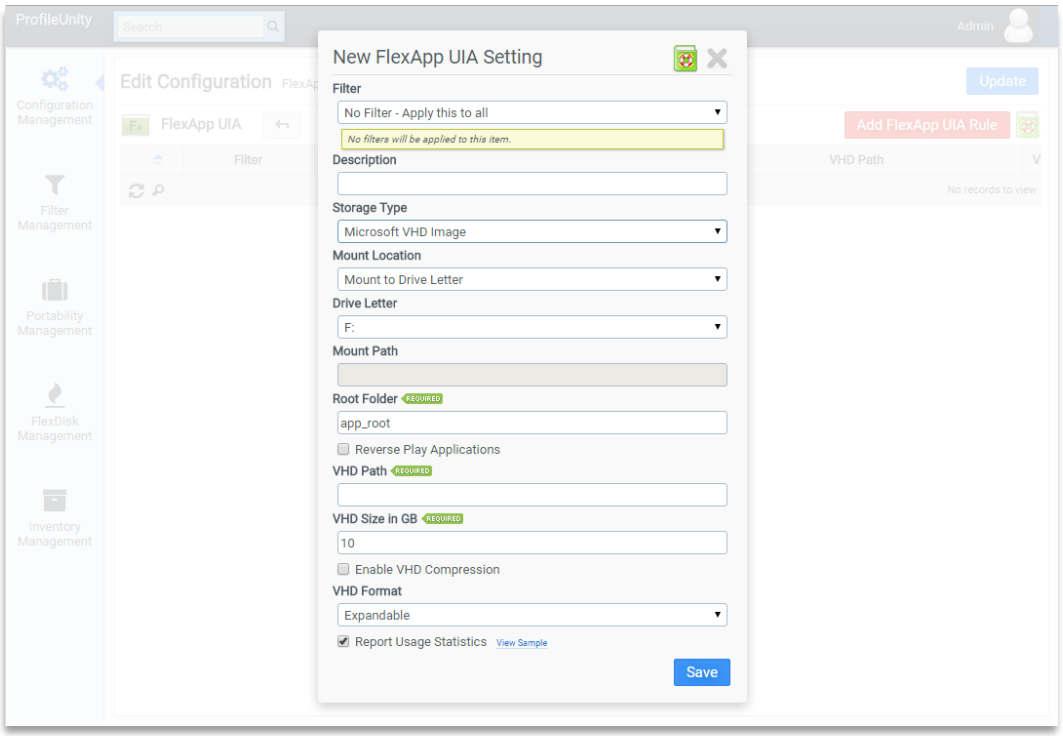
1. Select **Configuration Management** from the navigation tree on the left-hand side of the ProfileUnity Management Console.
2. Create a new configuration or choose the configuration you wish to edit and click on the **Edit** icon to the right of the configuration name.
3. To setup FlexApp UIA, click on the **FlexApp UIA** icon. *Since FlexApp UIA should not be used in conjunction with FlexApp DIA, the FlexApp UIA module is hidden by default. To enable the icon for use in Configuration Management, navigate to Administration > Settings > Miscellaneous and disable/uncheck the "Hide FlexApp UIA Module" option.*



4. Click on **Add a FlexApp UIA Rule** button.

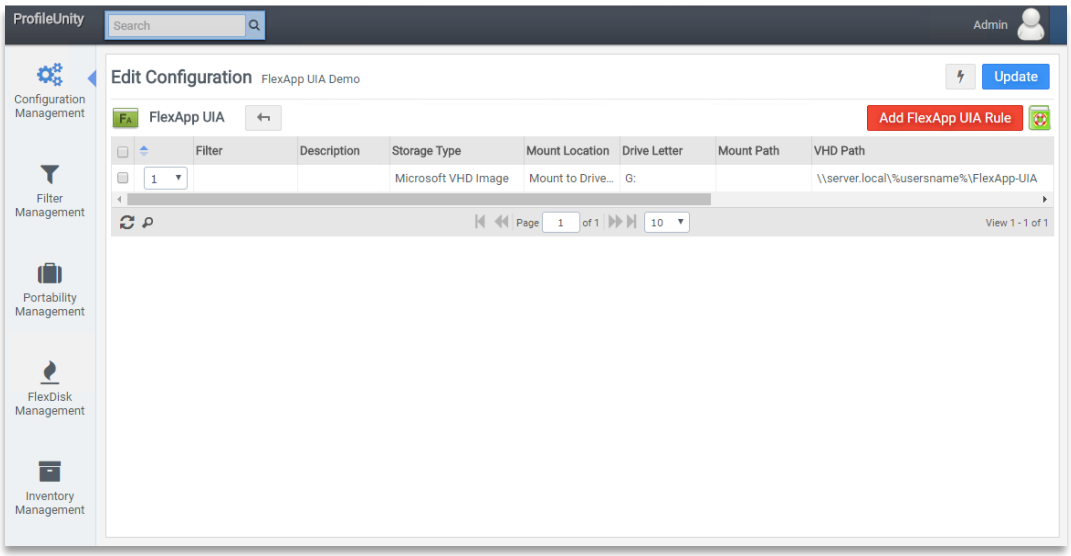


5. Optionally select a **Filter** for this configuration element.

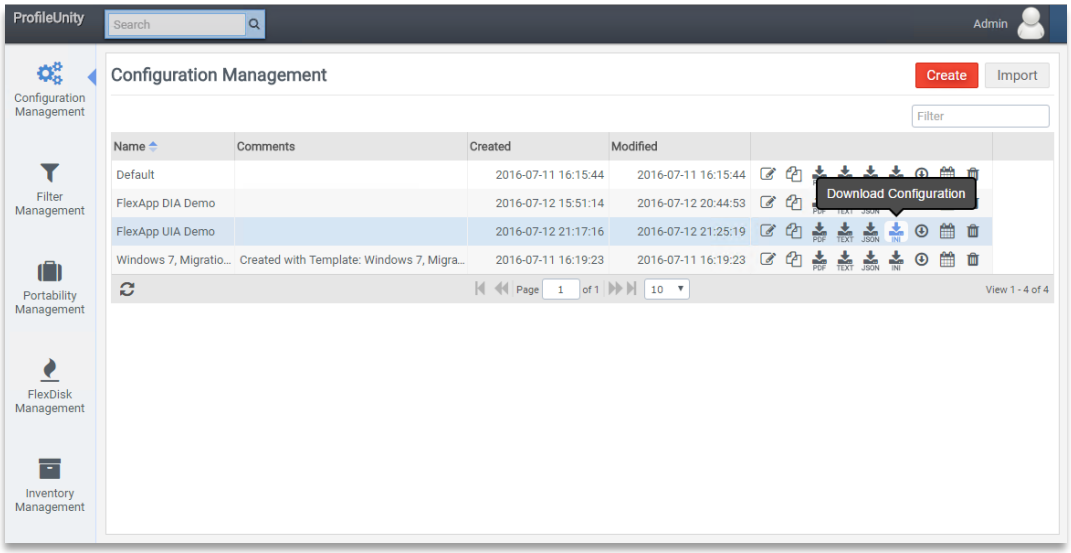


a. Choose the **Storage Type** and assign a **Drive Letter** to it.
b. Enter a **Root Folder**, **VHD Path,** and **VHD Size in GB**.
c. Click **Save**.

6. Click on the blue **Update** button in the upper right-hand corner to update the configuration.

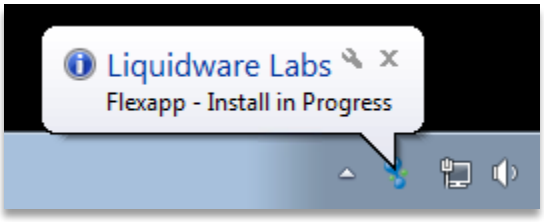


7. To activate your changes, remember to download a new INI file after you have made all changes to your configuration.



Installing Your First UIA Application

1. Inside your user session, download a copy of your application, for example Firefox.
2. Save it to disk.
3. Double click the installer (DO NOT RUN IT FROM THE BROWSER!).
4. If everything is working, you should see the following message in the notification area of the Taskbar during the install.

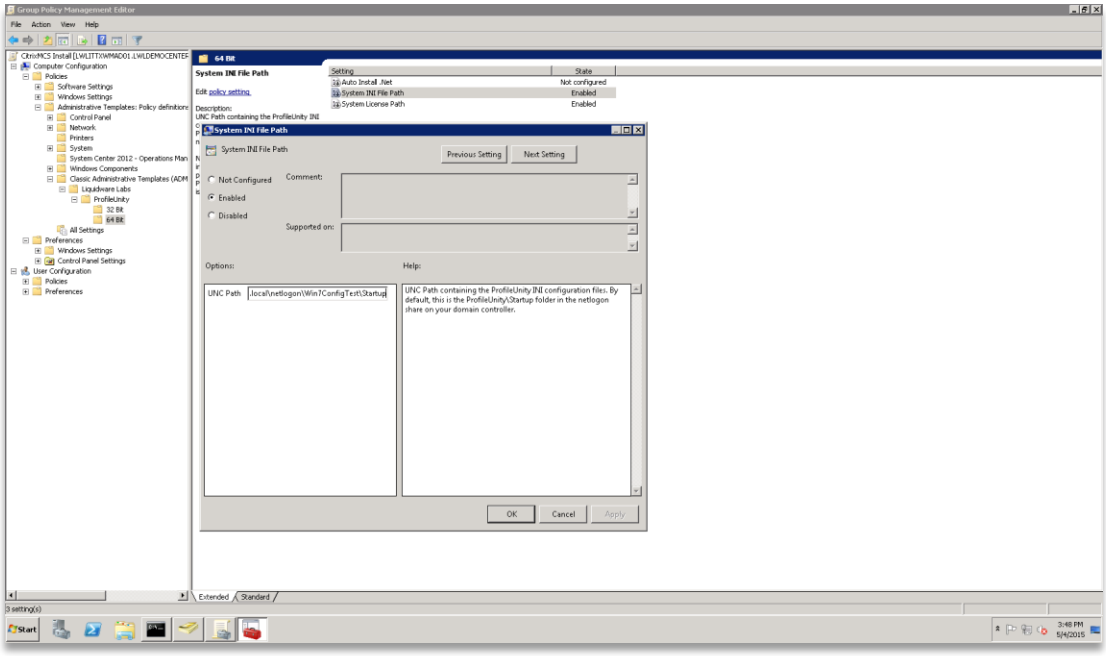


5. Also, on the next login you should still have the application you previously installed, which in this case was Firefox, as well.

How to Setup a VHD for Application Layering at the OS Level

Follow these directions to use a VHD at the OS level for assigning applications at computer startup with FlexApp or if you need to layer applications on RDSH or XenApp.

1. You will want to setup ProfileUnity to run at the system level. Start by creating a GPO on the computer OU where you want ProfileUnity to run on computer startup.
2. Setup startup.exe to run on computer startup.
3. Under Computer Configuration import the ProfileUnity ADM from the **Deployment Path**.
4. Then go to Classic Administrative Templates\Liquidware Labs\ProfileUnity\<Bit level of your desktops>
5. Set your System INI path to a directory under your **Deployment Path** called \Startup
    a. `\\dc.local\netlogon\ProfileUnity\Startup`

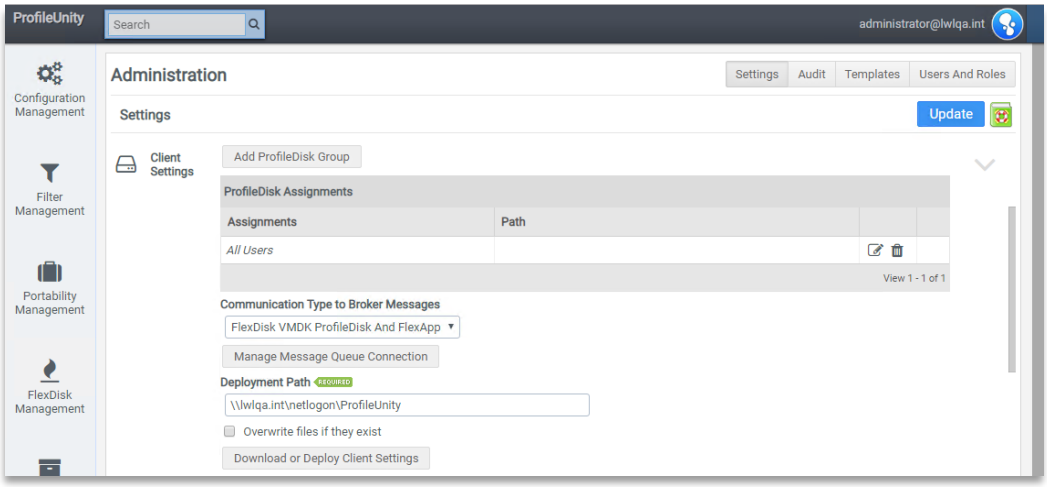


6. From the ProfileUnity Management Console, navigate to Administration > Settings > ProfileUnity Tools and go to **Run Client Tools As Service**. Fill in fields as seen including **domain\user** and **password**. This user DOES NOT need to be an administrator but does have to be a domain user account.
7. Click the blue **Update** button.
8. From the ProfileUnity Tools section, click the **Download or Deploy Service Configuration** button. The Deployment Path should be:
    a. `\\dc.local\netlogon\ProfileUnity`
9. Using the FlexApp Packaging Console, create a package selecting "VHD" as the **Package Type** and choose which datastore to which the package will be saved.
10. In the ProfileUnity Management Console, use the FlexApp DIA Configuration Module to assign FlexApp packages to users. Build a configuration with apps that will be global to the computer, then place that INI into \\dc.local\netlogon\ProfileUnity\Startup, nothing should in \Startup but an INI of things you want to run on computer startup.
11. Reboot or refresh all user desktops. If Microsoft® .NET Framework 4.6.2 is not installed in the base operating system on a particular desktop, ProfileUnity will install it once the desktop restarts. The installation of .NET 4.6.2 will increase the time it takes before the desktop is available again.

How to Setup a VMDK for Application Layering at the OS Level

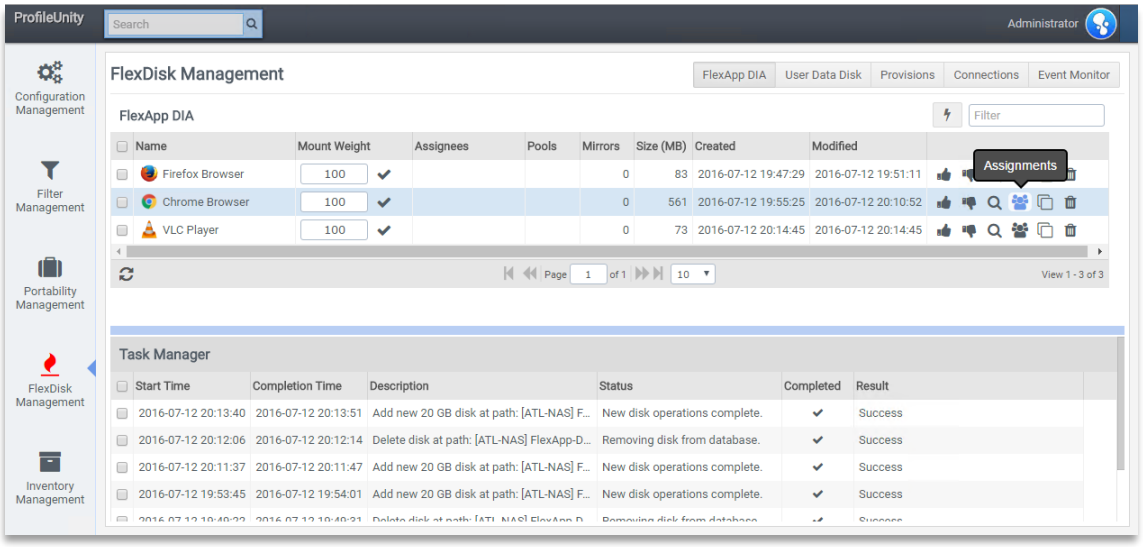
Follow these directions to use a VMDK at the OS level for assigning applications at computer startup with FlexApp or if you need to layer applications on RDSH or XenApp. These instructions are based on starting and using ProfileUnity as a Service in Step 13 that is needed for user or group assignments set in Step 20. FlexDisk also offers computer-level assignment of VMDK-based FlexApp packages that can be used without ProfileUnity as a Service. If ProfileUnity as a Service is used, FlexApp packages are played back on service boot making them available before the user logs in. If ProfileUnity as a Service is not used, playback of the FlexApp packages occurs at user login. However, no user assignments are needed in this case which may be valuable in a lab scenario where many users have access to login to a finite pool of computers.

1. When installing the ProfileUnity Management Console, you were asked to pick a broker for the integration of the FlexDisk VMDK distribution technology. FlexDisk will govern the operation of the VMDK ProfileDisk whether you use **Citrix**, **VMware**, **Microsoft WVD** or **Other** as your broker.
2. Login to the ProfileUnity Management Console and go to the Administration area. To get there, go to your user name at the top right of the Management Console and select **Administration** from the drop-down list.
3. If you selected VMware as your broker, you will need to setup the FlexDisk service if you haven't already done so. Please follow the directions given in the **How to Setup ProfileUnity Clustering for FlexDisk** section earlier.
4. In the Settings tab, scroll to the Client Settings section in the Administration area and:

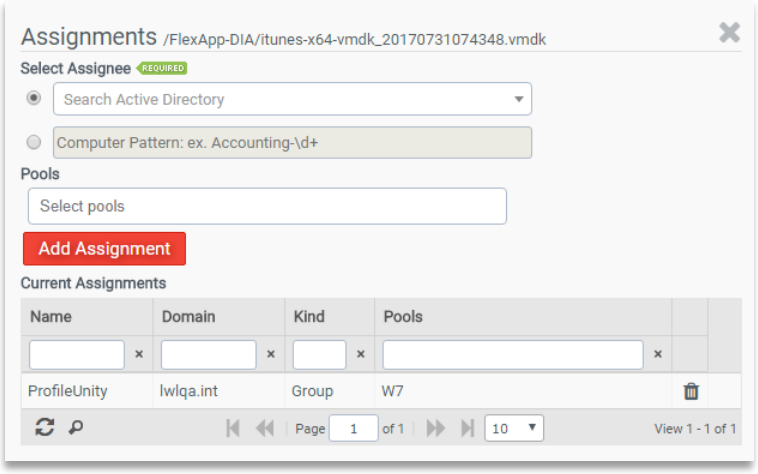


   a. Set **Communication Type to Broker Messages** to "FlexDisk VMDK ProfileDisk And FlexApp".
   b. Set the **Deployment Path** to your NETLOGON path.
   c. Click the blue **Update** button in the upper right corner of the Management Console.
   d. Click on the **Download or Deploy Client Settings** button to send a copy of the client configuration file to the specified location. Then you will be able to select your **Platform** for deployment. Choose "Domain" or "Cloud" to send the client configuration file to the specified **Deployment Path**. If not logged in as an Active Directory user, choose "Download" and manually place the client configuration file, clientsettings.xml, in the **Deployment Path**.
5. Scroll down to the ProfileUnity Tools section and click on **Download or Deploy Client Tools** to copy the ProfileUnity Client files to the NETLOGON share on your domain controller.
6. Go to the Configuration Management area and create a configuration file. In order for the FlexDisk Service to work, you must have at least one ProfileUnity Configuration File (INI) downloaded to your **Deployment Path**. In order to configure FlexDisk without other configuration settings, just create a blank configuration file in the Configuration Wizard by choosing a "Manual" configuration with no other settings in it.

7. Now you want to setup ProfileUnity to run at the system level. Start by creating a GPO on the computer OU where you want ProfileUnity to run on computer startup.
8. Setup startup.exe to run on computer startup.
9. Under Computer Configuration Import the ProfileUnity ADM from the **Deployment Path**.
10. Then go to Classic Administrative Templates\ Liquidware Labs\ProfileUnity\<Bit level of your desktops>
11. Set your System INI path to a directory under your **Deployment Path** called \Startup
    a. `\\dc.local\netlogon\ProfileUnity\Startup`
12. From the ProfileUnity Management Console, navigate to Administration > Settings > ProfileUnity Tools and go to **Run Client Tools As Service**. Fill in fields as seen including **domain\user** and **password**. This user DOES NOT need to be an administrator but does have to be a domain user account.
13. Click **Update** button.
14. From the ProfileUnity Tools section, click the **Download or Deploy Service Configuration** button. The deployment path should be:
    a. `\\dc.local\netlogon\ProfileUnity`
15. Using the FlexApp Packaging Console, create a package selecting “FlexDisk” as the **Package Type** and choose which FlexDisk datastore to which the package will be saved.
16. In the ProfileUnity Management Console, instead of using the FlexApp DIA Configuration Module click on **FlexDisk Management** on the left navigation panel. Click on the **FlexApp DIA** tab and then click on the **Assignments** icon to create an assignment for users or groups.



17. Select the user that was filled in for Step 13. If you have VMware View and have installed the Connection Service Monitor, you will get the option to limit the application to the user, group and/or the pool. Click on **Add Assignment** to add it to the list of current assignments. When done, click on the 'X'.



18. In the Management Console, build a configuration with apps that will be global to the computer, then place that INI into `\\dc.local\netlogon\ProfileUnity\Startup`, nothing should in `\Startup` but an INI of things you want to run on computer startup.
19. Reboot or refresh all user desktops. If Microsoft® .NET Framework 4.6.2 is not installed in the base operating system on a particular desktop, ProfileUnity will install it once the desktop restarts. The installation of .NET 4.6.2 will increase the time it takes before the desktop is available again.

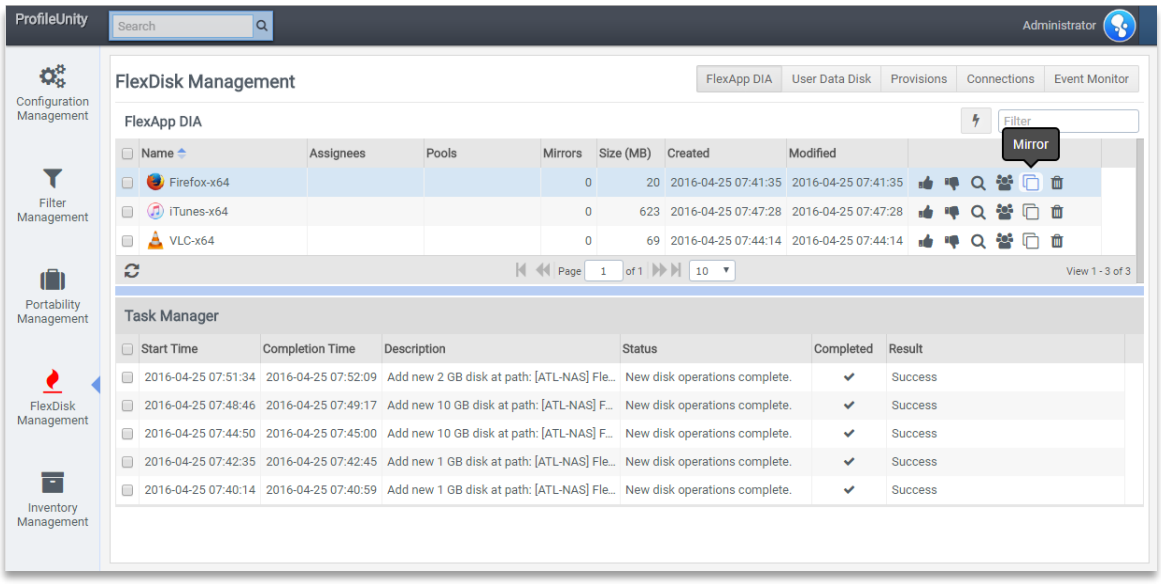
FlexApp: Mirroring VMDK-based Layers Across vCenter Servers and Storage Volumes

When your environment is made up of multiple vCenter Servers and/or multiple storage volumes, a problem arises with how to get a FlexApp Layer from one virtual center to another or from one storage volume A to another storage volume B. Cloning the application layer and using it separately from the original layer adds yet another layer of management overhead. FlexDisk allows mirroring of these VMDK-based FlexApp layers across vCenters or datastores without requiring additional layer management. With FlexApp Mirroring, administrators also have the ability to throttle the bandwidth so that network connections are not compromised.

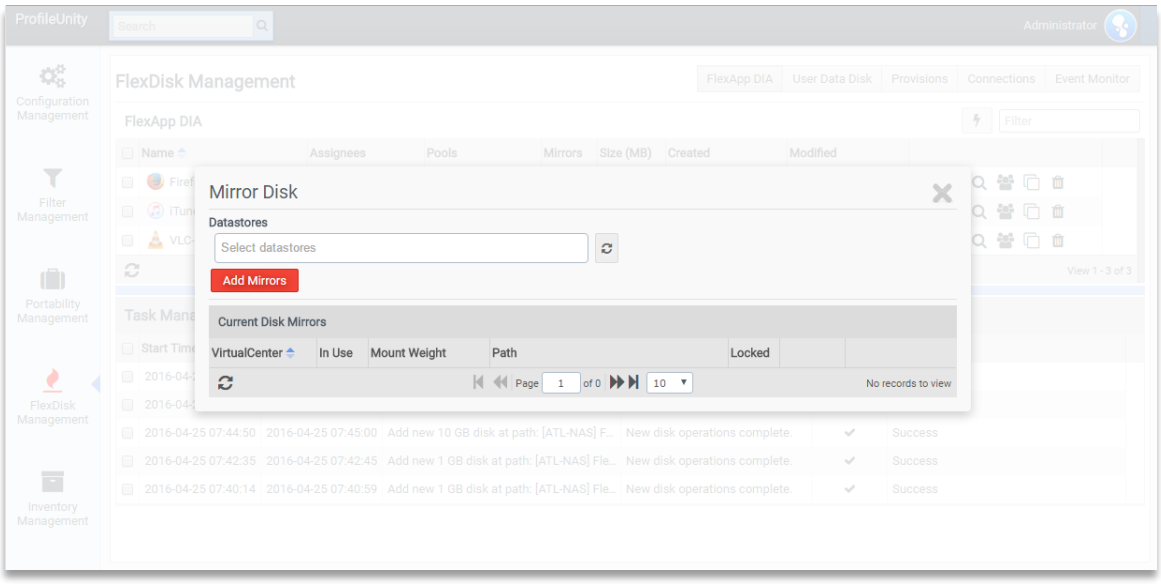
> ***Note that FlexApp Layers have an additional bit-level requirement. The bit-level of the operating system on which you package applications with the FlexApp Packaging Console needs to match the operating system of the desktop to which you deploy applications. If you are packaging applications to 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. This also applies to mirroring FlexApp Layers to other datastores or vCenters. The target environment needs to match the environment in which the FlexApp Layer was packaged.***

To configure VMDK-based FlexApp Mirroring:

1. In the ProfileUnity Management Console, go to the FlexDisk Management area.
2. Click on the **Mirror** icon to the right of the VMDK-base FlexApp Layer you want to use across another datastore or another Virtual Center.



3. Find your datastore or Virtual Center and click **Add Mirrors**.



Getting Help with ProfileUnity

If you have questions or run into issues while using 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.

Troubleshooting with the Software

ProfileUnity with FlexApp provides full logging capabilities to track activities. Once you have tried to duplicate the issue with full logging turned on, the logfile details can be used to help pinpoint the source of the problem you are experiencing. To turn logging on, go to your login user ID at the top right of the Management Console interface and select **Administration**. At the top of the Administration area, select **Settings**. As you scroll through the list you will see the ProfileUnity Console Service Log category. Set the **Level of Logging** to **Debug** to provide the most information. To view the log file, click on **View Log**. If you are still experiencing issues and need to contact technical support for additional help, the logfile can be sent to support for further evaluation.

Not sure about your configuration settings? Don't forget that ProfileUnity offers summary reports for all your configurations, filters, and portability settings. Simply click on **Report** next to the name of the configuration, filter, or portability setting for which you need a report. You can download and save the report in either a Portable Document Format (PDF) or text format. These summary reports can also be very helpful in troubleshooting issues.

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 online help included with your Liquidware Product.
- 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

Acknowledgements

ProfileUnity is partially built on the KiXtart scripting language developed by Mr. Ruud van Velsen. Liquidware Labs, Inc. would like to extend its heartfelt gratitude to Mr. Ruud van Velsen for developing and allowing us to incorporate KiXtart into the ProfileUnity product.

ProfileUnity uses AES encryption and decryption routines written by Dr. Brian Gladman. The source code for these routines is available to any interested party from the AES project page on Dr. Gladman's web site. This is the same AES code that is incorporated into ProfileUnity. Liquidware Labs, Inc. would like to express its appreciation to Dr. Gladman for making this code available.

Appendix A - Filter Rule Conditions & Examples

Filters give administrators the added flexibility of conditionally applying configuration settings. Filters allow you to provision resources based on a variety of conditions such as user group or user name. For example, you may want to map drives to users based on the department in which they work. ProfileUnity also offers the ability to create context-aware filters that provision resources based on the location of your user.

You will remember from our discussion on filters in the **Utilizing Filter Management** section in this manual that filters can have multiple filter rules. These filter rules are made up of a **Condition**, **Match**, and **Value** triplet. The following lists the conditions to choose from and some examples of how to use them:

Filter Rule Condition	Description
User Group Membership	Tests for user membership in specified user group. *Example:* • *Condition: User Group Membership* • *Match: Is (Exactly)* • *Value: Accounting* *Returns true for users that are members of the Accounting group.*
User Primary Group	Tests value against user's assigned Primary Group. *Example:* • *Condition: User Primary Group* • *Match: Is (Exactly)* • *Value: Domain Users* *Returns true for users with Domain Users assigned as the Primary Group.*
Machine Group Membership	Tests for membership in the specified machine group. *Example:* • *Condition: Machine Group Membership* • *Match: Is (Exactly)* • *Value: Accounting* *Returns true for machines that are members of the Accounting group.*
Username	Tests value against user's logon name. *Example:* • *Condition: Username* • *Match: Is (Exactly)* • *Value: GPBurdell* *Returns true if the user's logon name is GPBurdell.*
IP Address	Tests value against the IP address assigned to the client machine. If multiple IP addresses are assigned to the client, the first four IP addresses are tested. *Example:* • *Condition: IP Address* • *Match: Is (Exactly)*

Filter Rule Condition	Description
	• *Value: 192.168.1.1* *Returns true if the client machine has IP Address 192.168.1.1.* *Example:* • *Condition: IP Address* • *Match: Begins With* • *Value: 10.1* *Returns true if the first two octets of the client IP Address is 10.1.* *Example:* • *Condition: IP Address* • *Match: Range* • *Value: 192.168.1.1-192.168.1.10* *Returns true if the client machine has IP Address within the range of 192.168.1.1 through 192.168.1.10.*
Host Name	Tests value against the fully qualified TCP/IP hostname. *Example:* • *Condition: Host Name* • *Match: Is (Exactly)* • *Value: atl-001.xyz.com* *Returns true if the client machine has atl-001.xyz.com assigned as the TCP/IP hostname.*
Computer Name	Tests value against the NetBIOS computer name. *Example:* • *Condition: Computer Name* • *Match: Is (Exactly)* • *Value: ATL-001* *Returns true if the client machine is named ATL-001.*
MAC Address	Tests value against the address of the network adapter. *Example:* • *Condition: MAC Address* • *Match: Is (Exactly)* • *Value: 00306E053085* *Returns true if the client's network adapter has address 00306E053085.* *Example:* • *Condition: MAC Address* • *Match: Begins With* • *Value: 00306E* *Returns true if the client's network adapter has an address that begins with 00306E. Since the first six digits of a MAC address identify the vendor, this filter would return true for network adapters manufactured by Hewlett Packard.*

Filter Rule Condition	Description
Domain	Tests value against the domain the client machine is a member of. *Example:* • *Condition: Domain* • *Match: Is (Exactly)* • *Value: XYZ* *Returns true if the client machine is a member of the domain or workgroup XYZ.*
Logon Domain	Tests value against the domain the user is logged on to. *Example:* • *Condition: Logon Domain* • *Match: Is (Exactly)* • *Value: ABC* *Returns true if the user is logged on to domain ABC.*
Logon Server	Tests value against the name of the logon server. *Example:* • *Condition: Logon Server* • *Match: Is (Exactly)* • *Value: ADC01* *Returns true if the server named ADC01 processed the user's logon.*
Site	Tests value against the name of the Active Directory site the client machine resides in. *Example:* • *Condition: Site* • *Match: Is (Exactly)* • *Value: Default-First-Site-Name* *Returns true if the client machine resides in site Default-First-Site-Name.*
TS Client Name	Tests value against the NetBIOS name of the client machine connecting to the terminal server. *Example:* • *Condition: TS Client Name* • *Match: Is (Exactly)* • *Value: ATL-001* *Returns true if the client machine connecting to the terminal server is named ATL-001.*
TS Session Name	Tests value against the session name assigned to the client machine connecting to the terminal server. *Example:* • *Condition: TS Session Name* • *Match: Is (Exactly)*

Filter Rule Condition	Description
	• *Value: RDP-Tcp#48* *Returns true if the client machine connecting to the terminal server is assigned session name RDP-Tcp#48.* *Example:* • *Condition: TS Session Name* • *Match: Begins With* • *Value: RDP* *Returns true for all client machines connecting to the terminal server via the RDP protocol.* *Example:* • *Condition: TS Session Name* • *Match: Begins With* • *Value: ICA* *Returns true for all client machines connecting to the terminal server via the ICA protocol.*
Custom Function	(Deprecated feature) Allows filter logic to be extended through the use of a custom written function. Please see **Appendix D – Custom Functions for** further assistance.
OU User	Tests value against the name of the Active Directory OU the user's account resides in. *Example:* • *Condition: OU User* • *Match: Is (Exactly)* • *Value: OU=Test,DC=xyz,DC=com* *Returns true if the user's account resides in OU=Test,DC=xyz,DC=com.*
OU Computer	Tests value against the name of the Active Directory OU the client machine resides in. *Example:* • *Condition: OU Computer* • *Match: Contains* • *Value: OU=Atlanta* *Returns true if the OU the client machine resides in contains OU=Atlanta.*
View Client Name	Tests value against the name of the client machine connecting to the VMware View session. *Example:* • *Condition: View Client Name* • *Match: Is (Exactly)* • *Value: ATL-001* *Returns true if the client machine connecting to the VMware View session is named ATL-001.*

Filter Rule Condition	Description
OS Architecture	Tests value against OS Architecture. OS Architecture is 32 for 32-bit OS and 64 for 64-bit OS. *Example:* • *Condition: OS Architecture* • *Match: Is (Exactly)* • *Value: 32* *Returns true if the client machine is running 23-bit OS.*
View Client IP	Tests value against the IP address of the client machine connecting to the VMware View session. *Example:* • *Condition: View Client IP* • *Match: Begins With* • *Value: 192.168.0* *Returns true if the client machine connecting to the VMware View session has an IP address that begins with 192.168.0.*
View Client MAC	Tests value against the MAC address of the client machine connecting to the VMware View session. *Example:* • *Condition: View Client MAC* • *Match: IS (Exactly)* • *Value: 88-9F-FA-8E-63-EB* *Returns true if the client machine connecting to the VMware View session has MAC address 88-9F-FA-8E-63-EB.*
View Client Protocol	Tests value against the display protocol used by the client machine connecting to the VMware View session. *Example:* • *Condition: View Client Protocol* • *Match: IS (Exactly)* • *Value: PCOIP* *Returns true if the client machine connecting to the VMware View is using the PCOIP protocol.*
View Client Type	Tests value against the client type of the machine connecting to the VMware View session. *Example:* • *Condition: View Client Type* • *Match: IS (Exactly)* • *Value: Windows* *Returns true if the client machine connecting to the VMware View is running Windows.*

Filter Rule Condition	Description
View User Domain	Tests value against the domain name of the user connecting to the VMware View session. *Example:* • *Condition: View User Domain* • *Match: IS (Exactly)* • *Value: LWL* *Returns true if the domain name of the user connecting to the VMware View session is LWL.*
View User Name	Tests value against the user name of the user connecting to the VMware View session. *Example:* • *Condition: View User Name* • *Match: IS (Exactly)* • *Value: GBurdell* *Returns true if the user name of the user connecting to the VMware View session is GBurdell.*
View Machine Domain	Tests value against the domain name of the machine connecting to the VMware View session. *Example:* • *Condition: View Machine Domain* • *Match: IS (Exactly)* • *Value: EXAMPLE* *Returns true if the client machine connecting to the VMware View is a member of domain EXAMPLE.*
Environment Variable	Tests environment variables. This selection required multiple values. The | (pipe) character should be used as a separator. *Example:* • *Condition: Environment Variable* • *Match: Is (Exactly)* • *Value: test|1234* *Returns true if the environment variable %test% equals 1234.*
Registry Value	Tests registry values. This selection required multiple values. The | (pipe) character should be used as a separator. *Example:* • *Condition: Registry Value* • *Match: Is (Exactly)* • *Value: HKEY_CURRENT_USER\Software\Liquidware Labs\Example|1234* *Returns true if the registry value HKEY_CURRENT_USER\Software\Liquidware Labs\Example equals 1234.*

Filter Rule Condition	Description
Registry Value Exists	Tests if registry value exists. *Example:* • *Condition: Registry Value Exists* • *Match: Is (Exactly)* • *Value: HKEY_CURRENT_USER\Software\Acro Software Inc\CPW\Data1* *Returns true if the registry value HKEY_CURRENT_USER\Software\Acro Software Inc\CPW\Data1 exists.*
Registry Key Exists	Tests if registry key exists. *Example:* • *Condition: Registry Key Exists* • *Match: Is (Exactly)* • *Value: HKEY_CURRENT_USER\Software\Liquidware Labs\MyKey* *Returns true if the registry key HKEY_CURRENT_USER\Software\Liquidware Labs\MyKey exists.*
File	Tests file. This selection requires multiple values. The | (pipe) character should be used as a separator. *Example:* • *Condition: File* • *Match: Is (Exactly)* • *Value: C:|test.txt* *Returns true if a file with the name test.txt exists in the C:\ path.*
Directory	Tests directory. This selection requires multiple values. The | (pipe) character should be used as a separator. *Example:* • *Condition: Directory* • *Match: Is (Exactly)* • *Value: C:|windows* *Returns true if a directory with the name windows exists in the C:\ path.*
ICA Client Address	Tests ICA client address. *Example:* • *Condition: ICA Client Address* • *Match: Is (Exactly)* • *Value: 192.168.1.1* *Returns true if ICA client address is 192.168.1.1.*
ICA Client Name	Tests ICA client name. *Example:* • *Condition: ICA Client Name* • *Match: Is (Exactly)*

Filter Rule Condition	Description
	• *Value: MachineName-PC* *Returns true if ICA client name is MachineName-PC.*
Service Exists	Tests if service exists. *Example:* • *Condition: Service Exists* • *Match: Is (Exactly)* • *Value: SSDP Discovery* *Returns true if a service named SSDP Discovery exists.*
Service Running	Tests if service is running. *Example:* • *Condition: Service Running* • *Match: Is (Exactly)* • *Value: SSDP Discovery* *Returns true if a service named SSDP Discovery is running.*
Process Running	Tests if a process is running. *Example:* • *Condition: Process Running* • *Match: Is (Exactly)* • *Value: notepad.exe* *Returns true if a process named notepad.exe is running.*
Day of Week / Time of Day	(For use with English version Client only) Tests for a particular time block. Enter Value as a "Day|Time" pair separated by a vertical line or pipe. Use short day names: Sun, Mon, Tue, Wed, Thu, Fri, or Sat. The time should be specified using a 24-hour clock with hours from 0 to 23. *Example:* • *Condition: Day of Week / Time of Day* • *Match: Is (Exactly)* • *Value: Mon|9:30-10:00, Tue|9-18:31, Wed|12-13* *Returns true if Monday from 9:30am to 10:00am, Tuesday from 9:00am to 6:31pm, and Wednesday from 12:00pm to 1:00pm.*
LDAP Attribute	Tests for the value of an Active Directory attribute. Enter Value as an "Attribute|Value" pair separated by a vertical line or pipe. *Example:* • *Condition: LDAP Attribute* • *Match: Is (Exactly)* • *Value: department|engineering* *Returns true if the AD value of the department attribute is set to engineering.*

Appendix B – Configuration Macros

Macros can be used in your configuration anywhere an expression is expected. During logon, these macros expand into values based on their definition. For example, when a user with an ID of GPBurdell logs on to a client machine, the "@USERID" macro text is replaced with "GPBurdell". Macros have many practical uses. The UNC path, `\\your-server\@USERID`, can be used to map a drive to a per-user share. The @LSERVER macro can be used to set the system time on legacy clients from the authenticating domain controller. This table lists the supported macros and their definitions.

Macro	Definition
@ADDRESS	Address of the network adapter
@BUILD	Build number of the operating system
@COLOR	Current console color setting
@COMMENT	User comment
@CPU	Name of the central processing unit (CPU) (e.g.: "Intel Pentium III")
@CRLF	Carriage-return + Line-feed
@CSD	Reflects the most recent service pack version or CSD version information (e.g.: "Service Pack 1")
@CURDIR	Current directory
@DATE	Date (in the format YYYY/MM/DD)
@DAY	Day of the week (Monday, Tuesday, and so on)
@DOMAIN	Domain or workgroup the computer belongs to
@DOS	Version of Windows NT
@ERROR	Return code of the most recent command or function. A return code of 0 means the command or function was successful. Any other value indicates an error.
@FULLNAME	Full name of current user
@HOMEDIR	Short name of the directory part of home directory
@HOMEDRIVE	Drive letter of drive containing home directory
@HOMESHR	Server and share name part of home directory
@HOSTNAME	Fully qualified TCP/IP host name (including TCP/IP domain name)
@INWIN	Operating system: 1 = Windows NT; 2 = Windows 9x

Macro	Definition
@IPADDRESSx	TCP/IP address (possible values for x are 0 - 3)
@KIX	KiXtart product name and version
@LANROOT	Directory where network software resides (usually Systemroot\System32)
@LDOMAIN	Logon domain
@LDRIVE	Drive that is redirected to \\logonserver\NETLOGON
@LM	Version of network software
@LOGONMODE	If 1, indicates that KiXtart assumes to be running during the logon sequence
@LONGHOMEDIR	Long name of the directory part of home directory
@LSERVER	Logon server
@MAXPWAGE	Maximum password age
@MDAYNO	Day of the month as a number from 1 to 31
@MHZ	Approximation of the CPU speed. Not available on Windows 9x.
@MONTHNO	Month of the year as a number from 1 to 12
@MONTH	Name of the month (January, February, etc.)
@MSECS	Milliseconds part of the current time
@PID	Process ID of the KiXtart process
@PRIMARYGROUP	Current user's primary group
@PRIV	User's privilege level (GUEST, USER, ADMIN)
@PRODUCTSUITE	Operating system suite. Combination of any of the following values: 1 - "Small Business" 2 - "Enterprise" 4 - "BackOffice" 8 - "CommunicationServer" 16 - "Terminal Server" 32 - "Small Business (Restricted)" 64 - "EmbeddedNT" 128 - "DataCenter" 256 - "Single user Terminal Server" 512 - "Home Edition" 1024 - "Blade Server" 2048 - "Embedded (Restricted)"

Macro	Definition
	4096 - "Security Appliance" 8192 - "Storage Server" 16384- "Computer Cluster Server”
@PRODUCTTYPE	Operating system type. Possible values: <u>Windows Server 2003:</u> "Windows Server 2003" "Windows Server 2003 Domain Controller" "Windows Server 2003 R2" "Windows Server 2003 R2 Domain Controller" <u>Windows Server 2008:</u> "Windows Server 2008 (R2)" "Windows Server 2008 (R2) Core" "Windows Server 2008 (R2) Small Business Edition" "Windows Server 2008 (R2) Enterprise Edition" "Windows Server 2008 (R2) Enterprise Edition Core" "Windows Server 2008 (R2) Datacenter Edition" "Windows Server 2008 (R2) Datacenter Edition Core" "Windows Server 2008 (R2) Enterprise Edition for Itanium" "Windows Server 2008 (R2) Web Server Edition" "Windows Server 2008 (R2) Web Server Edition Core" "Windows Server 2008 (R2) Compute Cluster Edition" "Windows Server 2008 (R2) Home Edition" "Windows Storage Server 2008 (R2) Express Edition" "Windows Storage Server 2008 (R2) Standard Edition" "Windows Storage Server 2008 (R2) Enterprise Edition" "Windows Storage Server 2008 (R2) Small Business Edition" "Windows Server 2008 (R2) Essential Business Server" "Windows Server 2008 (R2) Essential Business Server Premium" "Windows Server 2008 (R2) Essential Business Server Management" "Windows Server 2008 (R2) Essential Business Messaging" "Windows Server 2008 (R2) Essential Business Security" "Windows Server 2008 (R2) Hyper-V" "Windows Server 2008 (R2) Foundation" <u>Windows XP:</u> "Windows XP Home Edition" "Windows XP Professional" "Windows XP Professional Tablet PC" "Windows XP Media Center Edition" "Windows XP Starter Edition" “Windows Fundamentals for Legacy PCs” <u>Windows Vista:</u> "Windows Vista Starter Edition" "Windows Vista Home Basic Edition" "Windows Vista Home Basic Edition N" "Windows Vista Home Premium Edition"

Macro	Definition
	"Windows Vista Business Edition" "Windows Vista Business Edition N" "Windows Vista Enterprise Edition" "Windows Vista Ultimate Edition" <u>Windows 7:</u> "Windows 7 Starter Edition" "Windows 7 Starter Edition N" "Windows 7 Home Basic Edition" "Windows 7 Home Basic Edition N" "Windows 7 Home Premium Edition" "Windows 7 Home Premium Edition N" "Windows 7 Business Edition" "Windows 7 Business Edition N" "Windows 7 Professional Edition" "Windows 7 Professional Edition N" "Windows 7 Enterprise Edition" "Windows 7 Enterprise Edition N" "Windows 7 Ultimate Edition" "Windows 7 Ultimate Edition N"
@PWAGE	Password age
@RAS	Number of active Remote Access Service (RAS) connections
@RESULT	Returns command specific information (e.g.: the drive letter of an automatic redirection command)
@RSERVER	KXRPC server used for the current session
@SCRIPTDIR	Directory of current script
@SCRIPTEXE	Name of KiXtart executable (”KIX32.EXE”, ”WKIX32.EXE”)
@SCRIPTNAME	Name of current script
@SERROR	Error text string corresponding with @ERROR
@SID	Current user's Windows NT Security Identifier (SID)
@SITE	Name of the site in which the system resides
@STARTDIR	Directory from which KiXtart was started
@SYSLANG	Full English name of the language of the operating system specified in the format defined by ISO Standard 639 (example : ”0413Dutch (Standard)”)
@TICKS	Returns the number of milliseconds that have elapsed since the system was started

Macro	Definition
@TIME	Current time (in the format HH:MM:SS)
@USERID	Current user's Windows NT user ID
@USERLANG	Full English name of the language selected by the current user specified in the format defined by ISO Standard 639 (example : ”0413Dutch (Standard)”)
@WDAYNO	Day of the week as a number from 1 to 7. Monday has a value of 1.
@WKSTA	Computer name
@WUSERID	Current user's Windows user ID
@YDAYNO	Day of the calendar year as a number from 1 to 365 starting with January 1
@YEAR	Current year

Appendix C - Configuration Modules

A configuration is comprised of settings within one or more configuration modules. Each configuration module is designed to accomplish a specific task.

Using Macros

Macros can be used in your configuration anywhere an expression is expected. During logon, these macros expand into values based on their definition. For example, when a user with an ID of GPBurdell logs on to a client machine, the "@USERID" macro text is replaced with "GPBurdell". Macros have many practical uses. Please see **Appendix B – Configuration Macros** for a list of supported macros and their definitions.

Using Reserved Characters

The characters @, %, and $ are reserved characters. If you want to use these characters in your configuration, you will need to use @@, %%, or $$ instead. Using two reserved characters in a row acts as an escape sequence which allows you to use the character in some other capacity rather than the function for which it was reserved. For example, if you want to use the email address `user@domain.com`, you will need to use `user@@domain.com` instead.

The following configuration modules are available:

Configuration Module	Description
Main	Defines parameters that apply to all users
Privilege Elevation	Allows non-admin users to install applications that have been approved by the Allow and Deny policy rules defined by the administrator
Application Restrictions	Allows or denies users access to applications
FlexApp UIA	Controls the storage and retrieval of user installed applications
FlexApp DIA	Controls the delivery of department installed applications
Virtual Disks	Used to mount storage
Printers*	Connects and disconnects network printers **Starting with version 6.7, this module was moved from following the Path module to now following the Virtual Disks module which will affect the order in which the module settings are processed. Registry rules will now be processed earlier.*
Drive Mapping*	Connects and disconnects network drives **Starting with version 6.7, this module was moved from following the Desktop Start Menu module to now preceding the Portability Settings*

Configuration Module	Description
	module which will affect the order in which the module settings are processed. Registry rules will now be processed earlier.
Portability Settings	Controls the storage and retrieval of personal user preferences during logoff and logon
File Associations	Allows administrators to control what apps open for what file types
Profile Cleanup	Used to clean up a user's profile contents
User Defined Aliases	Retrieves information from your domain controllers about the current user's account for use throughout your ProfileUnity configuration
User Defined Scripts	Allows a custom written KiXtart script to execute either before or after ProfileUnity runs
Application Launcher	Launches applications on your client machines during or after ProfileUnity execution
Desktop Start Menus	Allows a user to add a start menu to the desktop
Environment Variables	Sets and clears environment variables
Registry*	Makes changes to the registry on client machines **Starting with version 6.5, this module was moved from following the RDP Client module to now following the Environment Variables module which will affect the order in which the module settings are processed. Registry rules will now be processed earlier.*
Administrative Templates	Allows ADMX import with GPO import direct from Active Directory
Folder Redirection	Redirects shell folders to an alternate location, and optionally move or copy existing data
INI Files	Enables ProfileUnity to manipulate INI files during user logon
Internet Explorer	Configures Internet Explorer features such as Start Page, Download Directory, and Favorites for Internet Explorer versions 4.0 and higher
Internet Proxy	Adjusts Internet Explorer's proxy settings to match your network configuration for Internet Explorer versions 4.0 and higher
Inventory	Creates a report listing computer name, computer domain membership, username, processor, memory, display resolution, MAC address, IP address, operating system (including service pack version and installed hotfixes), Internet Explorer version, and installed applications
MAPI Profiles	Configures Windows Messaging subsystem profiles used by Microsoft Outlook for access to Microsoft Exchange server

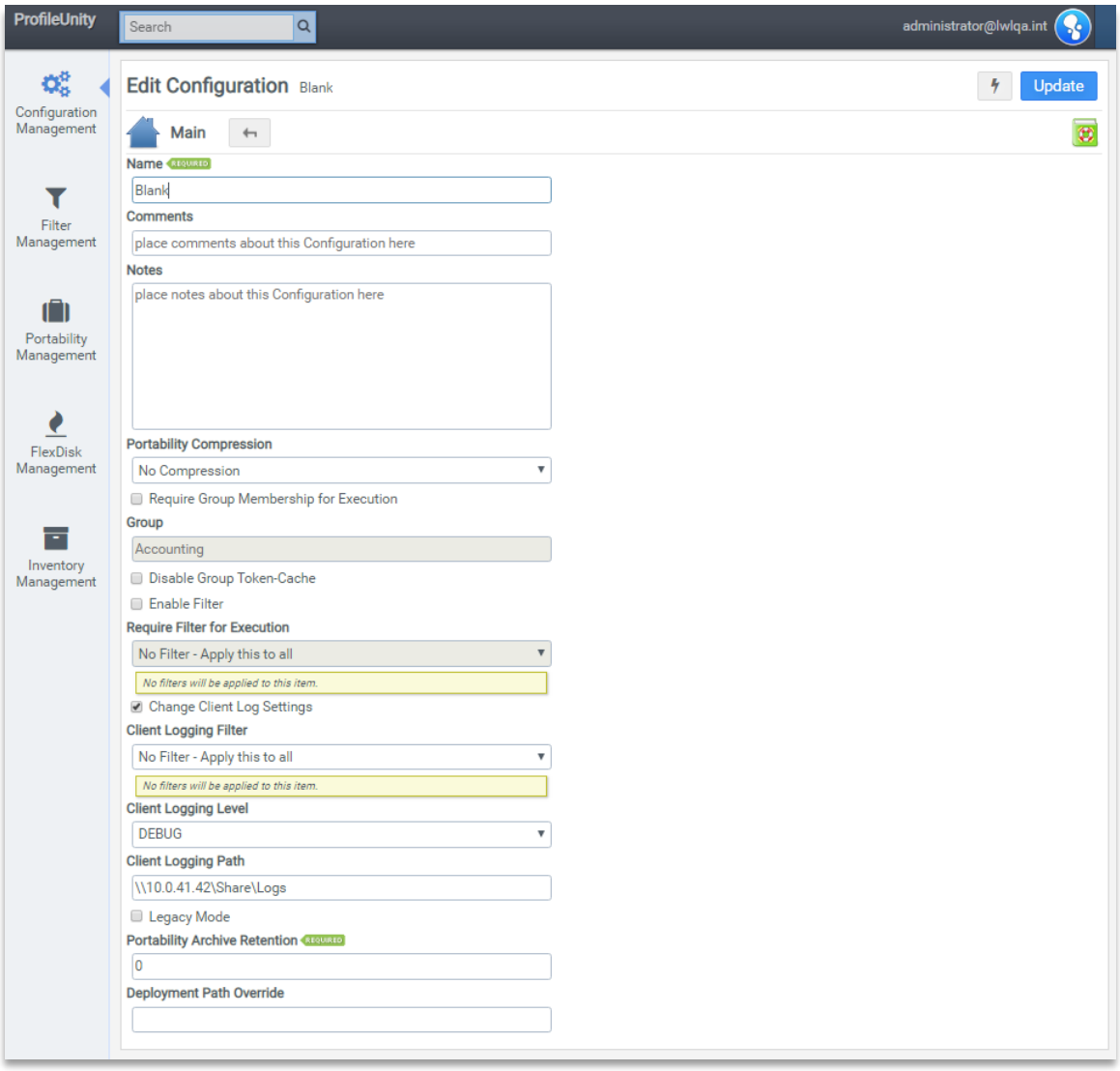
Configuration Module	Description
Message Boxes	Displays popup messages to your users during logon
Office File Locations	Modifies the default file locations for Microsoft Office
Office Options	Configures Microsoft Office user information and sets options that control interaction with the user
Outlook	Modifies settings that control Microsoft Outlook's behavior for message handling, message format, and AutoArchive
Paths	Modifies the search path
RDP Client	Configures RDP clients
Shortcuts	Creates shell shortcuts and web links
ThinApp	Dynamically provision ThinApp packages
Time Sync	Sets the time on your client machines from a centralized source
Trigger Points	Allows other modules to run based on various events that occur on the machine
Windows Options	Modifies settings that control the Windows user interface

NOTE: Most configuration modules are processed in the order they are listed above. This can impact the expected results. For example, since the Environment Variables module is processed after the Application Launcher module, the Application Launcher module will not have access to any necessary environment variables set by the Environment Variables module.

However, you can layer your configuration settings by using multiple configurations. If any configuration named "Default" exists, it is run first by ProfileUnity. Then the remaining configurations are run in alphabetical order. Configurations that are processed later in the sequence can take advantage of settings that were processed earlier. For example, to use an environment variable in the Application Launcher module, you could configure the Environment Variables in the Default configuration and then have a separate configuration that uses the Application Launcher module. In this case, the Default configuration would run first and set up the environment variables. When the Default configuration finishes running, the remaining configurations would be processed in alphabetical order allowing the Application Launcher module to take advantage of the environment variable settings.

Main

The Main configuration module is used to define parameters for a configuration that apply to all users.



Name:
Enter a name that helps you identify your configuration.

> ***When ProfileUnity executes, it obtains the list of available configurations from the authenticating domain controller. If a configuration named Default is present, it is always processed first. The remaining configurations are processed in alphabetical order.***

Comments:
Enter a description that helps you identify your configuration.

Notes:

Enter Notes that help you identify your configuration.

Portability Compression:
Choose the compression algorithm used by the Portability Settings configuration module:

- **Smaller Archives** – uses LZMA compression for smaller archives
- **Faster Archives** – uses LZO compression for faster performance
- **No Compression**

Require Group Membership for Execution:
Selecting this option restricts execution of this configuration to members of the global group specified in the **Group** field.

Group:
Only available if **Require Group Membership for Execution** is selected

Disable Group Token-Cache:
ProfileUnity caches group membership information in the registry. Caching the group membership information reduces the amount of network traffic generated during user logon. However, if an existing group is renamed, the token-cache will not immediately update itself. If you experience problems filtering by Group Membership, selecting this option will disable the cache.

Enable Filter:
Check this option to restrict the execution of this configuration based on a filter.

Require Filter for Execution:
Only available if **Enable Filter** is selected. The execution of this configuration will be restricted to members of the filter specified such as operating system, Microsoft Office version detected, or any custom filter.

Change Client Log Filter:
Check this option to restrict the logging of this configuration based on a filter.

Client Logging Filter:
Only available if **Enable Client Log Filter** is selected. The logging of this configuration will be restricted to members of the filter specified such as operating system, Microsoft Office version detected, or any custom filter.

Client Logging Level:
Only available if **Enable Client Log Filter** is selected. The logging level controls the amount of logging information output. Choose from: **Fatal or Debug.**

Client Logging Path:
Only available if **Enable Client Log Filter** is selected. Choose where to store the log file.

Legacy Mode:
Affects how the Portability Compression algorithms will work. In order to use older compression algorithms that will support Windows Vista, or Windows Server 2008 users, please turn on Legacy Mode by selecting the checkbox. If the checkbox is not selected, the ProfileUnity Client will use newer compression technology intended for Windows 7 and above users.

NOTE: If an operating system prior to Windows 7 is detected when attempting to use the new compression technology, the ProfileUnity Client will fall back to the older compression method which is slower.

Portability Archive Retention:
The maximum number of copies of user portability settings to keep. A user's portability settings are saved at session logoff. Copies of archives and manifests will be saved to a Retention folder under where user settings are stored. Any or all of the copies may be used as a restore point to restore a user's settings to a previous state. The default is 0 which will not retain any archives.

Deployment Path Override:
Use this setting to override the default deployment path set in the **ProfileUnity Tools** section of the **Administration > Settings** tab. Leave this blank to use the default deployment path from the **Administration** area. Note that in order for this override setting in the Main configuration module to work, the **Overwrite files if they exist** option in the **ProfileUnity Tools** section of the admin settings must be checked. The overwrite option is a global setting in the management console. If overwrites are not permitted (checked), then the **Deployment Path Override** will be ignored.

Privilege Elevation

The Privilege Elevation module allows standard users to securely install and run applications needing elevated rights without making the user an administrator. The Allow and Deny policy rules defined by the administrator determine how the privileges are applied to users. Privilege Elevation applies to Windows 7 and higher or Windows Server 2008 R2 and higher machines only.

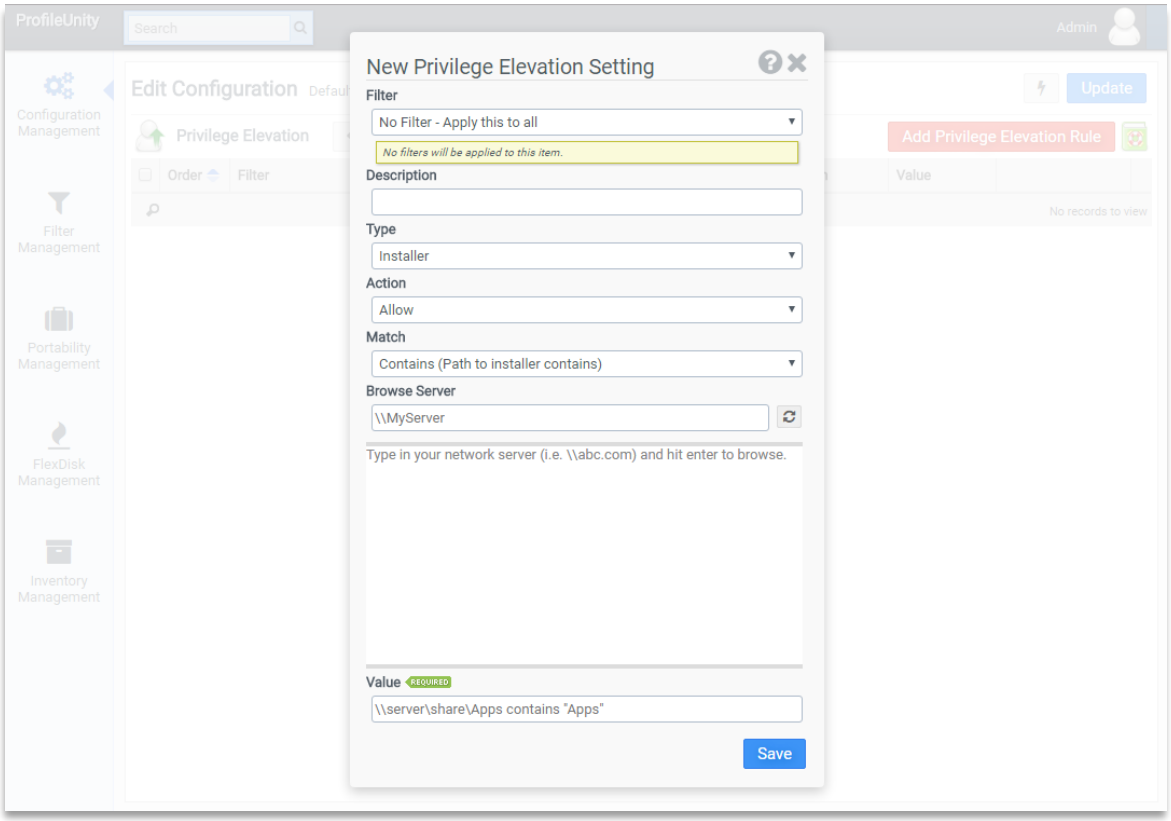
Use this module along with the **Application Restrictions** & **FlexApp UIA** modules to provide Application Rights Management (ARM) that enables administrators to securely grant specific users detailed application rights without making them a Windows Administrator.

The Privilege Elevation module allows for several different scenarios:

1. If FlexApp UIA is enabled for a user and the Privilege Elevation rule type equals **Installer**, ProfileUnity monitors user installed applications in non-persistent virtual environments granting rights based on the rule.

2. If FlexApp UIA is disabled for a user and the Privilege Elevation rule type equals **Installer**, ProfileUnity either allows or denies users permission to install applications on physical or persistent virtual desktops based on the rule.

3. If the Privilege Elevation rule type equals **Application**, ProfileUnity can securely elevate an application that requires administrative rights to run without making the user an administrator for their entire session.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Type:
Select whether to apply privilege elevation to an **Installer** or an **Application**.

Action:
Select one of the following actions:

- **Allow**
- **Deny**

Match:
Select the Match condition to test against. Choose from: **Contains, Equals, Hash, Starts With, Ends With,** or **Signed**.

Browse Server:
If the privilege elevation applies to an **Installer**, select **Browse Server** and then type your server name and press **Enter** to search for valid path names you can click on to use in the **Value** field rather than typing the full path name to the **Installer**.

Value:
Enter the value to test against.

Example:

Filter: No Filter – Apply this to all
Type: Application
Action: Allow
Match: Signed
Value: Liquidware Labs, Inc.

The above example will allow non-admin users to install applications signed by Liquidware Labs, Inc.

Application Restrictions

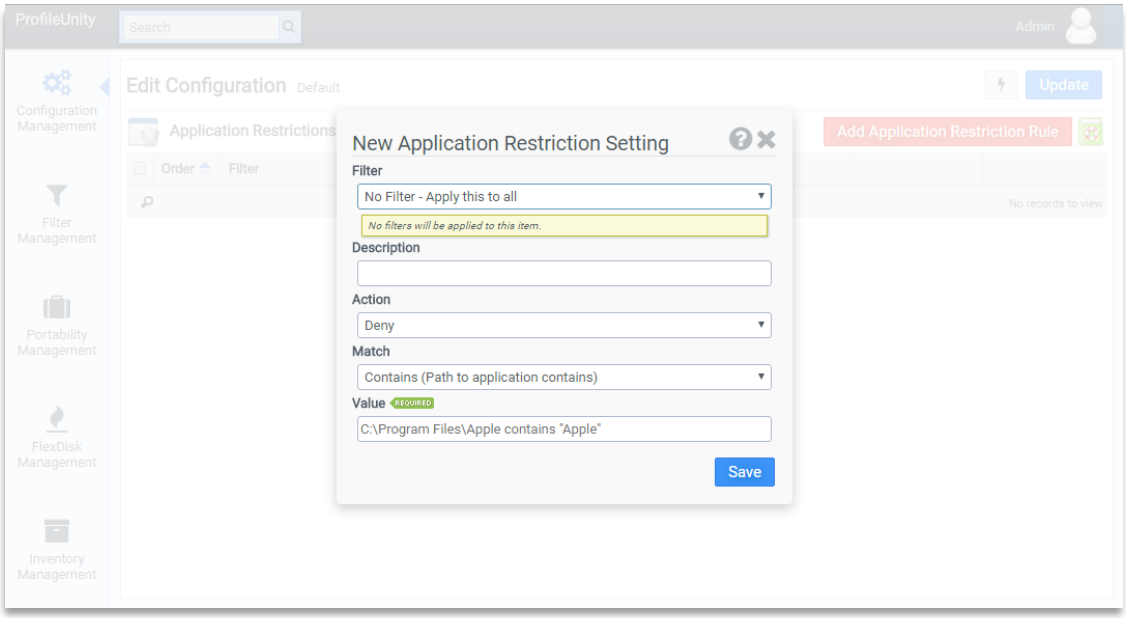
The Application Restrictions module allows or denies users access to applications providing allow/deny options for installed applications per user. Application Restrictions applies to Windows 7 and higher or Windows Server 2008 R2 and higher machines only.

Use this module along with the **Privilege Elevation** & **FlexApp UIA** modules to provide Application Rights Management (ARM) that enables administrators to securely grant specific users detailed application rights without making them a Windows Administrator. More specifically, the Application Restrictions module allows organizations to minimize their number of base images while ensuring they are compliant with licensing and imaging agreements.

Administrators can add any number of both **Allow** and **Deny** rules in the Application Restriction module. The rules are evaluated sequentially starting with the first one at the top of the list. When a filter returns true for an individual user, then that rule sets whether the Application Restriction module becomes an "all allow" or "all deny" list for that user. If that first true rule has an **Allow Action**, then any of the remaining **Allow** rules in the module list where the filter is also true will apply. If instead the first true filter rule has a **Deny Action**, then it will become an "all deny" list for that user.

The application access policy will be applied based on whether the rules evaluate to create an Allow List or a Deny List. If the rules evaluate to create a Deny List for a user, then the user can run any applications except for those specified in the deny rules. If the rules evaluate to create an Allow List for a user, then the user can only run applications that are specified by the allow rules. Access to other applications will be denied. By default, the following processes are automatically allowed:

- ProfileUnity processes
- All system processes
- Processes from C:\Windows and subdirectories
- Processes signed by VMware or Citrix



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Action:
Select one of the following actions:

- **Allow**
- **Deny**

Match:
Select the Match condition to test against. Choose from: **Contains, Equals, Hash, Starts With, Ends With,** or **Signed**.

Value:
Enter the value to test against.

Example:

Filter: No Filter – Apply this to all
Action: Allow
Match: Signed
Value: Liquidware Labs, Inc.

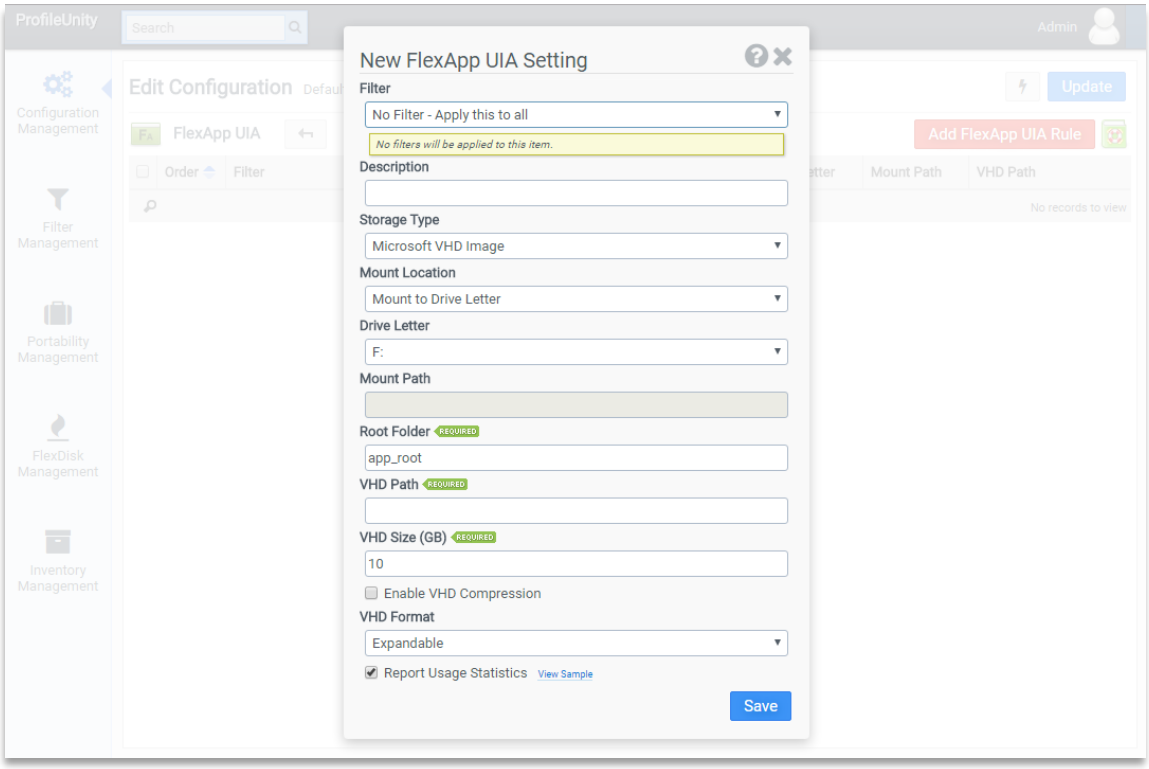
The above example will allow non-admin users to run applications signed by Liquidware Labs, Inc.

FlexApp UIA

The FlexApp UIA module is used to control the storage and retrieval of user installed applications in non-persistent desktop environments. Upon logout, ProfileUnity will automatically reverse play the application and remove the application from the system. FlexApp UIA applies to Windows 7/8.1 or Windows Server 2008 R2/2012 R2 only. To activate this module, check **Enable FlexApp UIA Module** in the Miscellaneous Settings section in the Administration area of the management console.

> ***Please note the following FlexApp UIA limitations. FlexApp UIA is not supported on Windows 10, Windows Server 2016, Windows Server 2019 or on persistent desktops with any OS. FlexApp UIA and FlexApp DIA should not be used together on a user's desktop. They can be used separately.***

Use this module along with the **Privilege Elevation** & **Application Restrictions** modules to provide Application Rights Management (ARM) that enables administrators to securely grant specific users detailed application rights without making them a Windows Administrator.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Storage Type:
Select the type of storage to use for user installed applications: **Microsoft VHD Image**

Mount Location:
Select the type of location that will be used to mount the user storage for any applications. Choose from **Mount to Drive Letter** or **Mount to Path**.

Drive Letter:
If **Mount to Drive Letter** is selected, choose the drive letter to use for user installed application storage. When using the VHD storage type, the VHD image will be mounted at the chosen drive letter. When using the persistent disk storage type, the chosen letter will need to match the letter where the persistent disk is already mounted.

Mount Path:
If **Mount to Path** is selected, enter the path to the user installed application storage.

Root Folder:
The **Root Folder** designates the top level or root folder used to store user installed applications on the selected **Drive Letter**.

VHD Path:
The VHD Path designates the UNC location ProfileUnity will use to store/retrieve the VHD file containing the user installed applications.

VHD Size in GB:
Enter the maximum size in gigabytes that the VHD will be allowed to use. This parameter is only used during initial VHD creation.

Enable VHD Compression:
Selecting this option will enable compression on the VHD volume. This option is only used during initial VHD creation.

VHD Format:
Select whether the VHD volume should be **Expandable** or **Fixed** size. This option is only used during initial VHD creation.

Report Usage Statistics:
Reports usage statistics back to Liquidware.

Example:

Filter: No Filter – Apply this to all
Storage Type: Microsoft VHD Image
Mount Location: Mount to Drive Letter
Drive Letter: F:
Root Folder: app_root
VHD Path: \\ATLFS02\@userid$
VHD Max Size: 10GB
Enable VHD Compression: Selected
VHD Type: Expandable
Report Usage Statistics: true

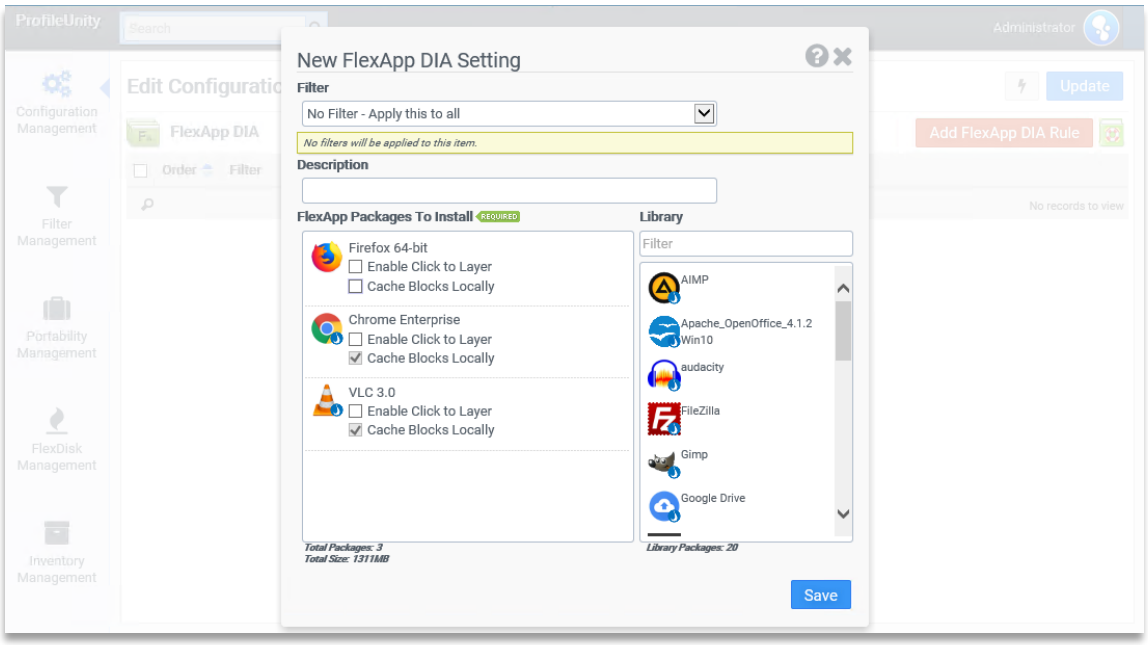
The above example will create a VHD named `flexapp.vhd` located in the `\\ATLFS02\@USERID` share. The VHD volume will be mounted as drive letter `F:`, limited to 10GB in size, have compression enabled, and will be expandable. User installed applications will use `F:\app_root` for storage and will be played back

after the desktop loads. This example leverages the macro `@USERID`. When ProfileUnity executes, the `@USERID` macro is expanded into the username. Usage statistics will report back to Liquidware.

FlexApp DIA

The FlexApp DIA module is used to control the storage and retrieval of department installed applications. Upon logout, ProfileUnity will automatically reverse play the application and remove the application from the system. FlexApp DIA applies to Windows 7 and higher or Windows Server 2008 R2 and higher machines only.

Please note that FlexApp UIA and FlexApp DIA should not be used together on a user's desktop. They can be used separately.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

FlexApp Packages to Install:
Drag packaged applications from the **Library** list over to this list so that they can be included with the user's profile at login.

Enable Click to Layer:
This enables the Click-to-Layer feature. Click-to-Layer postpones layering in a FlexApp package at login time while still giving the user an application shortcut. If and when the user clicks on the shortcut to use the application during their session, ProfileUnity layers in the application on demand. This feature is useful for applications that are less frequently used.

Note: If the desired application has been packaged prior to FlexApp Packaging Console v6.7, the package can be automatically updated for Click-to-Layer compatibility by opening the FlexApp Packaging Console version 6.7 or higher, selecting 'Edit the package's metadata' for the desired package and clicking OK.

Cache Blocks Locally:
Check **Cache Blocks Locally** if desired. FlexApp Packages can be configured to replay using a cached mode on user desktops which is particularly helpful with inconsistent network connections, SMB shares behind firewalls, and routers that have many hops. In cached mode, ProfileUnity will request blocks of the FlexApp package and store them locally so the next time the block is referenced, it is read from local storage. **Cache Blocks Locally** is enabled by default for FlexApps on cloud storage and is optional for FlexApps on SMB storage.

Library:
Displays applications that have been packaged by the FlexApp Packaging Console and are ready for deployment.

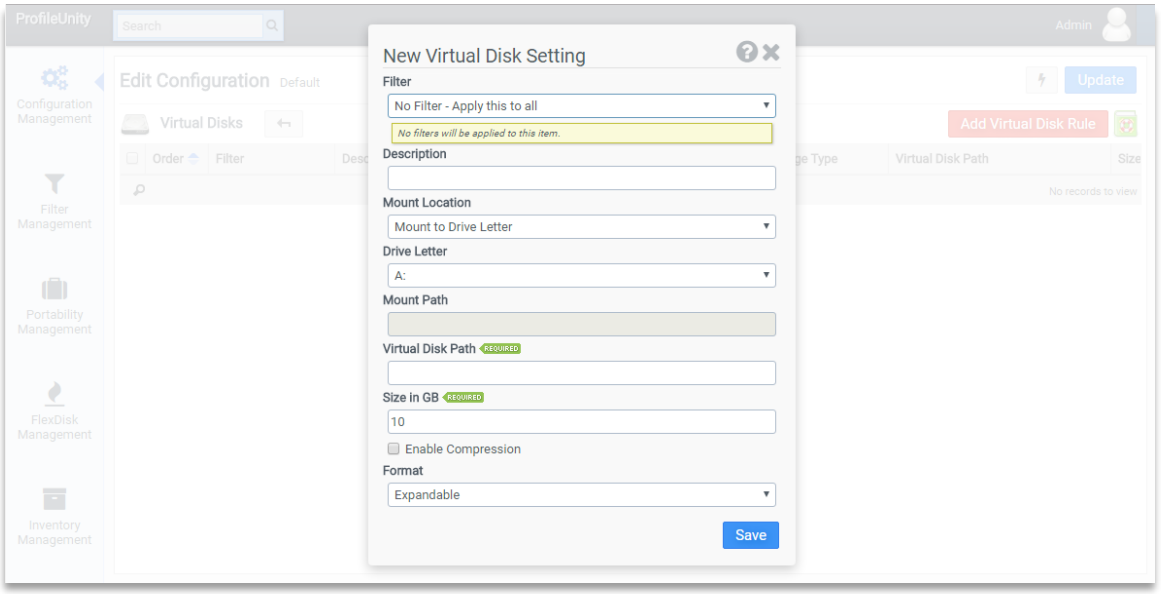
Example:

Filter: No Filter - Apply this to all
FlexApp Packages to Install: Firefox
Enable Click To Layer: Checked
Enable Cache Blocks Locally: Checked

The above example will install Firefox for all users and activate the "Click-To-Layer" feature. Blocks will be cached locally. Upon logoff, the application will be removed from the system.

Virtual Disks

The Virtual Disks module is used to mount storage. Virtual Disks applies to Windows 7 and higher or Windows Server 2008 R2 and higher machines only.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Mount Location:
Select where to mount the disk. Choose from:

- **Mount to Drive Letter**
- **Mount to Path**

Drive Letter:
Select the drive letter to which the image will be mounted. This option is only available if **Mount to Drive Letter** is chosen for **Mount Location**.

Mount Path:
Select the path to which the image will be mounted. This option is only available if **Mount to Path** is chosen for **Mount Location**.

Virtual Disk Path:
The Virtual Disk Path designates the UNC location ProfileUnity will use to store/retrieve the image file.

Size in GB:
Enter the maximum size in gigabytes that the Virtual Disk will be allowed to use. This parameter is only used during initial Virtual Disk creation.

Enable Compression:
Selecting this option will enable compression on the volume. This option is only used during initial Virtual Disk creation.

Format:
Select whether the virtual disk should be **Expandable** or **Fixed** size. This option is only used during initial Virtual Disk creation.

Example:

Filter: No Filter - Apply this to all
Mount Location: Mount to Drive Letter
Drive Letter: A:
Mount Path:
Virtual Disk Path: \\mydomain\vd\mydisk.vhd
Size in GB: 10
Enable Compression: Unchecked
Format: Fixed

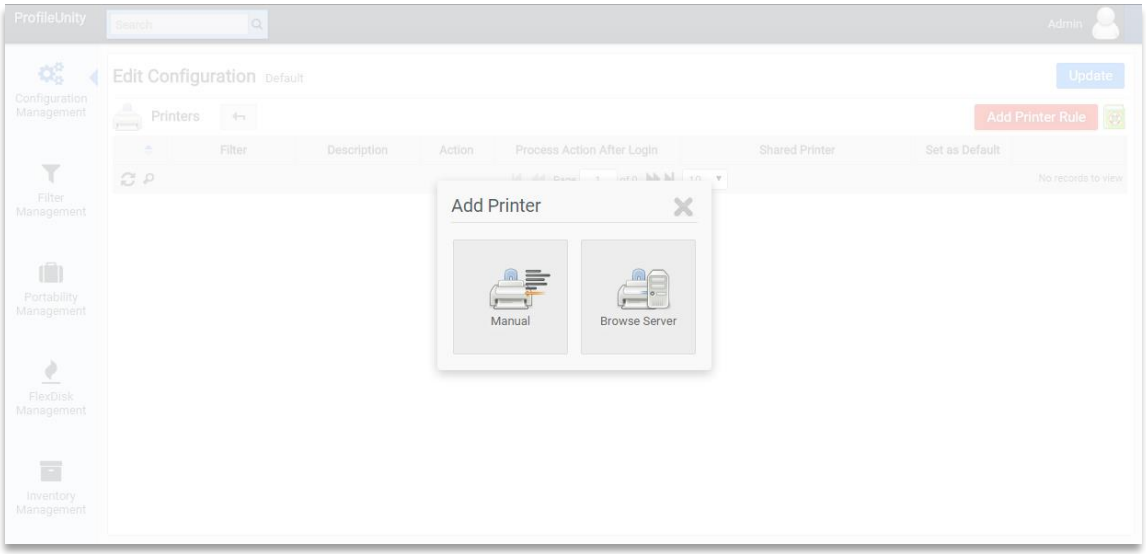
The above example will create a new VHD of fixed size 10GB. The disk will not be compressed when created. The disk will be placed in `\\mydomain\vd` and mounted to drive letter `A:`. The disk will be mounted for all users.

Printers

This module allows you to connect and disconnect network printers. Options include capturing a port, setting the printer as default, and automatically adding the printer. Filters are frequently used with the Printer module to manage printer assignments. Filters provide a wide-array of context-aware environment and security options. In a mobile environment, assigning printers can be based on the location of the desktop so that users are only assigned printers they can reasonably access. Printer access can also be restricted, for example, to certain users authorized to use special watermark paper on specific printers for deliverables such as checks, official transcripts or prescriptions.

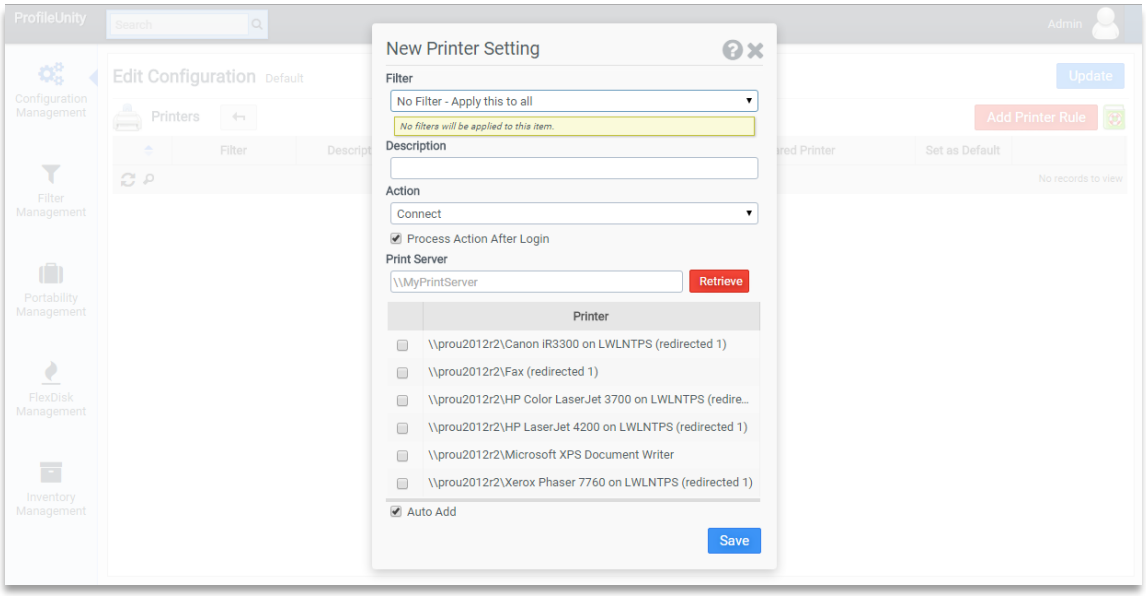
Note: This configuration module has moved. Formerly, the Printers module followed the Path module. Starting with 6.7, the Printers module now follows the Virtual Disks module. Since most module settings are processed in the same order as they appear in the Configuration Management area, it is possible for behavioral differences to exist between older and newer configurations created with different versions of the ProfileUnity Management Console.

Choose whether to add a printer manually or browse the print server to select multiple printers to be added at one time.

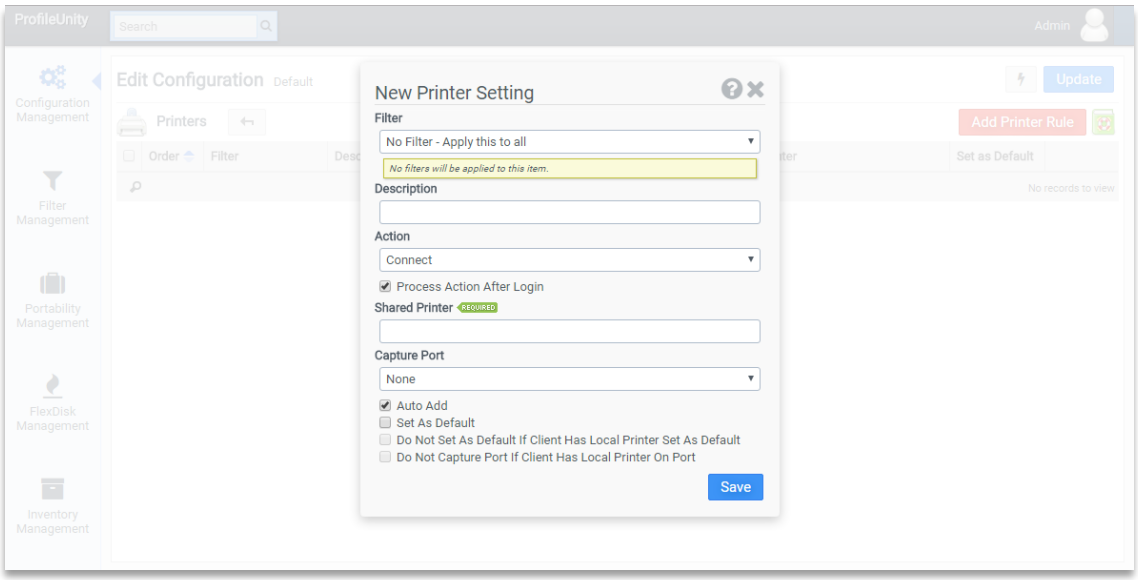


Select **Browse Server** to see a list of printer options.

Enter the path to the **Print Server**, and click the **Retrieve** button. Select the printers to add from the list of available printers on that server.



Select **Manual** to setup individual printers manually.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Action:
Select one of the following actions:

- **Connect** – Installs a network printer connection / captures a port.
- **Disconnect** – Removes a network printer connection / disconnects a port.
- **Disconnect All** – Removes all network printer connections and disconnects LPT ports 1-9.

Process Action After Login:
Determines whether the printer mapping should occur before or after the user has logged in.

Shared Printer:
Enter the UNC name of the shared printer. This option is not available if **Disconnect All** is selected as the **Action**.

NOTE: When Disconnect is selected as the Action, the server name portion of the UNC name must match the server name used when the printer was connected. For example, if the printer was connected with the UNC name `\\psvr.company.com\hplj`, you cannot use the UNC name `\\psvr\hplj` for the Disconnect Action.

Capture Port:
Select the port you want to capture or disconnect. This option is not available if **Disconnect All** is selected as the **Action**.

Auto Add:
Selecting this option will automatically install the printer on Windows clients. This option is only available if **Connect** is selected as the **Action**.

NOTE: This option leverages Microsoft Point and Print technology. The printer and drivers must be properly configured on the server for this option to function. The easiest way to test Point and Print is to attempt connecting to the printer from a client. If a client machine can connect without prompting for drivers, Point and Print is properly configured.

Set as Default:
Selecting this option will set the printer as the default. This option is only available if **Connect** is selected as the **Action** and **Auto Add** is selected.

Do Not Set as Default if Client has Local Printer Set as Default:
Selecting this option will prevent the printer from being set as the default if the client system has a local printer configured as the default. This option is only available if **Connect** is selected as the **Action** and **Set as Default** is selected.

Do Not Capture Port if Client has Local Printer on Port:
Selecting this option will prevent the port from being captured if the client system has a local printer installed on the port. This option is only available if **Connect** is selected as the **Action** and a **Capture Port** has been selected.

Example 1:

Filter: No Filter – Apply this to all
Action: Connect
Process Action After Login: Selected
Shared Printer: \\PSVR\NYCP07
Capture Port: LPT1:
Auto Add: Selected
Set as Default: Selected
Do Not Set as Default if Client has Local Printer Set as Default: Not Selected
Do Not Capture Port if Client has Local Printer on Port: Not Selected

Example 1 redirects the `LPT1:` port on all clients after the user logs in +to the network printer `NYCP07` on server `PSVR`. This example will also install the network printer and set it as the default printer on the client machines.

NOTE: When a network printer is installed, the drivers used by the printer are transferred from the server to the client system.

Example 2:

Filter: No Filter – Apply this to all
Action: Disconnect
Process Action After Login: Selected
Shared Printer: \\PSVR\ATLP01
Capture Port: Not Selected

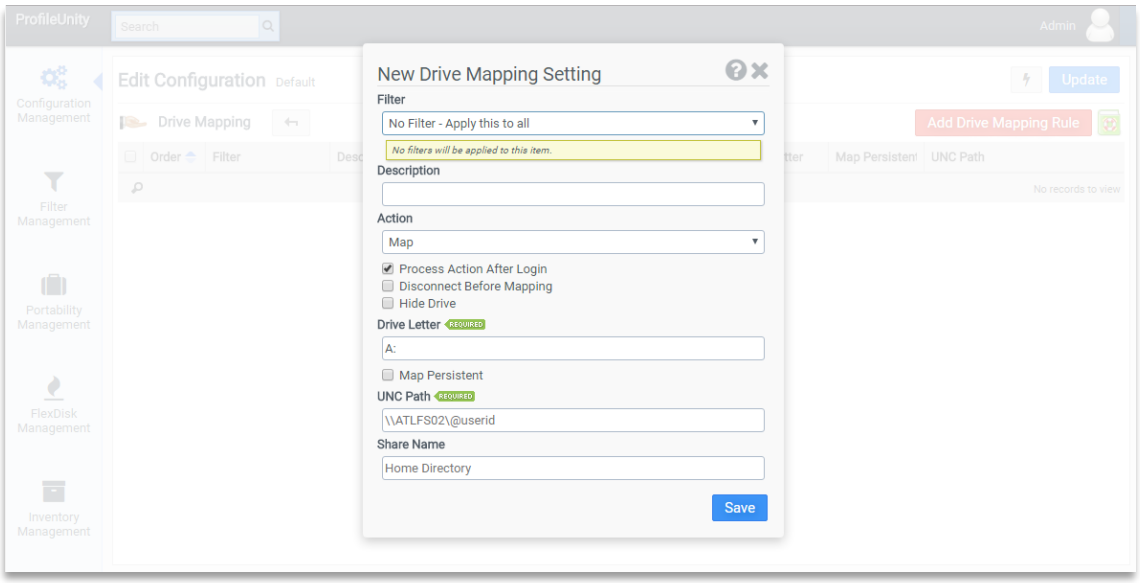
Example 2 removes the network printer `\\PSVR\ATLP01` after the user logs in on the client machines.

NOTE: When a network printer is removed, the drivers used by the printer are not removed from the client system.

Drive Mapping

The Drive Mapping module is used to connect and disconnect network drives.

Note: This configuration module has moved. Formerly, the Drive Mapping module followed the Desktop Start Menu module. Starting with 6.7, the Drive Mapping module now precedes the Portability Settings module. Since most module settings are processed in the same order as they appear in the Configuration Management area, it is possible for behavioral differences to exist between older and newer configurations created with different versions of the ProfileUnity Management Console.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Action:
Choose whether you want to **Map** a drive, **Disconnect** a drive, or **Disconnect All** drives.

Process Action After Login:
Determines whether the drive mapping should occur before or after the user has logged in.

Disconnect Before Mapping:
Selecting this option will disconnect the drive before mapping it. This option only applies if you choose **Map** for the action.

Hide Drive:
Selecting this option will hide the drive. This option only applies if you choose **Map** for the action.

Drive Letter:
Enter the drive letter you want to Map or Disconnect. This option does not apply if you choose **Disconnect All** for the Action.

Map Persistent:
Persistent drive mappings are mappings that are reconnected automatically at the next logon. This option only applies if you choose **Map** for the action.

UNC Path:
Enter the UNC Path you want the Drive Letter mapped to. This option only applies if you choose **Map** for the action.

Share Name:
The Explorer Label is the name that will show up in My Computer and the standard Windows dialog boxes next to the drive letter. This option only applies if you choose **Map** for the action.

NOTE: ProfileUnity leverages Windows Scripting Host to change the Explorer Label. If you disable Windows Scripting Host on your clients, ProfileUnity will be unable to modify the Explorer Label.

Example 1:

Filter: No Filter – Apply this to all
Action: Disconnect All
Process Action After Login: Selected

Example 1 will disconnect all drives after the user logs in.

Example 2:

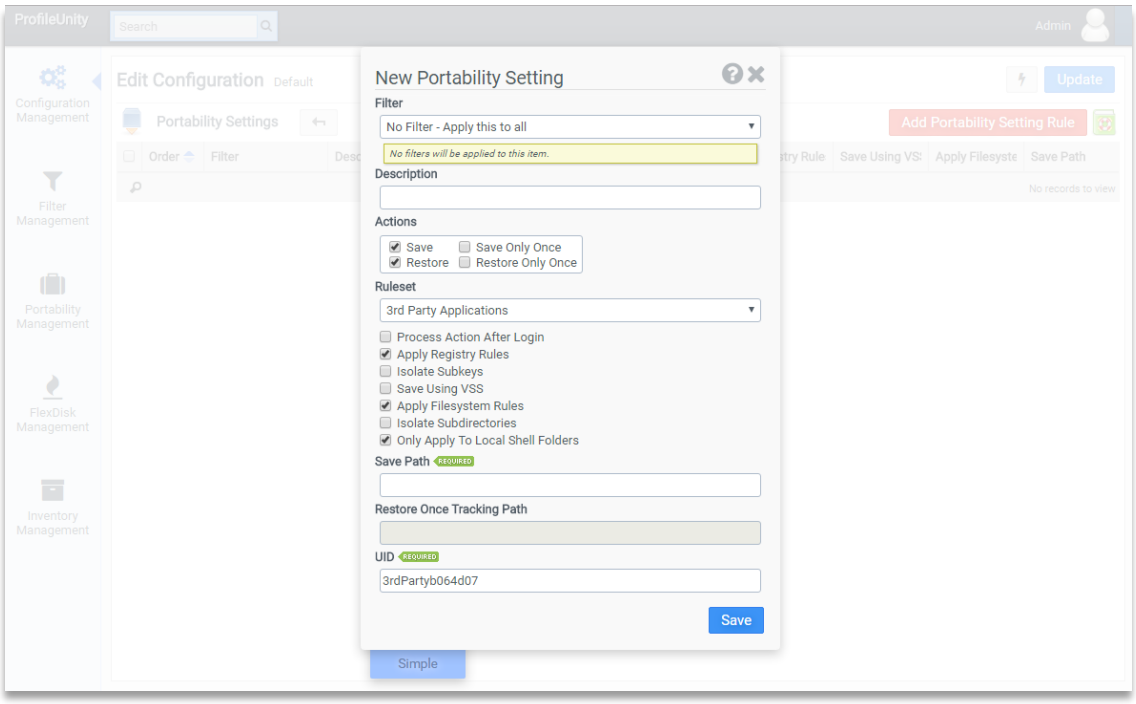
Filter: No Filter – Apply this to all
Action: Map
Process Action After Login: Selected
Drive Letter: H:
Map Persistent: Enabled
UNC Path: \\ATLFS02\@userid
Share Name: Home Directory

Example 2 leverages the macro @USERID. When ProfileUnity executes, the @USERID macro is expanded into the username. Therefore, the H: drive will be mapped to a per-user share located on server ATLFS02. The drive will be mapped persistent and will be assigned the label Home Directory after the user logs in.

Portability Settings

The Portability Settings module is used to control the storage and retrieval of personal user preferences during logoff and logon. This module works in conjunction with the rulesets defined through the Portability Management interface. Please see the **Providing Profile Portability & Management** section for assistance with Portability rulesets.

All options for the Advanced settings are shown below.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Actions:
Select one or more of the following actions:

- **Save** – During logoff settings are saved.
- **Save Only Once** – Available if you select **Save**. Settings are only saved once during logoff.
- **Restore** – During logon settings are restored.
- **Restore Only Once** – Available if you select **Restore**. Settings are only restored once during logon.

Ruleset:
Select the name of the ruleset you want assigned to this configuration element. This option does not apply if you choose **Restore** for the action.

Process Action After Login:
Indicates whether the action should occur before or after the user has logged in. Moving select rules to process after login can speed up login times. Do not select this option if data is needed immediately after login.

Apply Registry Rules:
Applies registry rules belonging to the ruleset.

Isolate Subkeys:
Isolate the subkeys based on the session.

Save Using VSS:
The Save Using VSS (also known as Volume Shadow Copy Service or Volume Snapshot Service) option in ProfileUnity's Portability engine allows ProfileUnity to backup files that could be locked for use or need quiescing before being saved. On logoff, a shadow copy is taken of the volume where all running processes will quiesce. Then the files listed in the portability rule are saved out. After the files are copied out, the shadow copy is removed. On single user desktops where the shadow copy services are set to disabled, ProfileUnity will set them to manual to perform the requested task then set back to disabled again. For RDS hosts, you need to have the volume shadow copy services set to manual.

Apply File System Rules:
Applies filesystem rules belonging to the ruleset.

Isolate Subdirectories:
Isolate the subdirectories based on the session.

Only Apply to Local Shell Folders:
Filesystem rules belonging to the ruleset will not run if the rule is for a shell folder that is already set to a mapped drive or folder share.

Save Path:
The **Save Path** designates the UNC name of the location ProfileUnity will use to store/retrieve the portability settings.

Restore Once Tracking Path:
The Restore Once Tracking Path designates the UNC name of the location ProfileUnity will use to save information about whether a restore has been performed. If blank, this information will be stored in the same directory as Save Path.

UID:
The **UID** is a unique identifier that is automatically generated by ProfileUnity. Normally, the **UID** does not need to be changed. The **UID** is used for file name generation.

ProfileUnity will create a compressed file named "[UID].7z" in the location specified by the Save Path. This file will be used by ProfileUnity to store/retrieve the portability settings.

Example 1:

Filter: No Filter – Apply this to all
Action: Save/Restore
Ruleset: MAPI Profiles
Apply Registry Rules: Enabled
Isolate Subkeys: False
Apply Filesystem Rules: Enabled
Isolate Subdirectories: False
Save Path: H:_Settings
Restore Once Tracking Path: blank
UID: v84lktv1tat6lan

Example1 saves the settings defined by the `MAPI Profiles` ruleset at logoff and restores them at logon. Registry rules and Filesystem rules defined in the ruleset will be applied. The settings will be stored and retrieved from a file named `v84lktv1tat6lan.7z` located in the `H:\_Settings` folder.

NOTE: This example assumes the H: drive is mapped to a network share. It is also possible to use a UNC name for the Save Path.

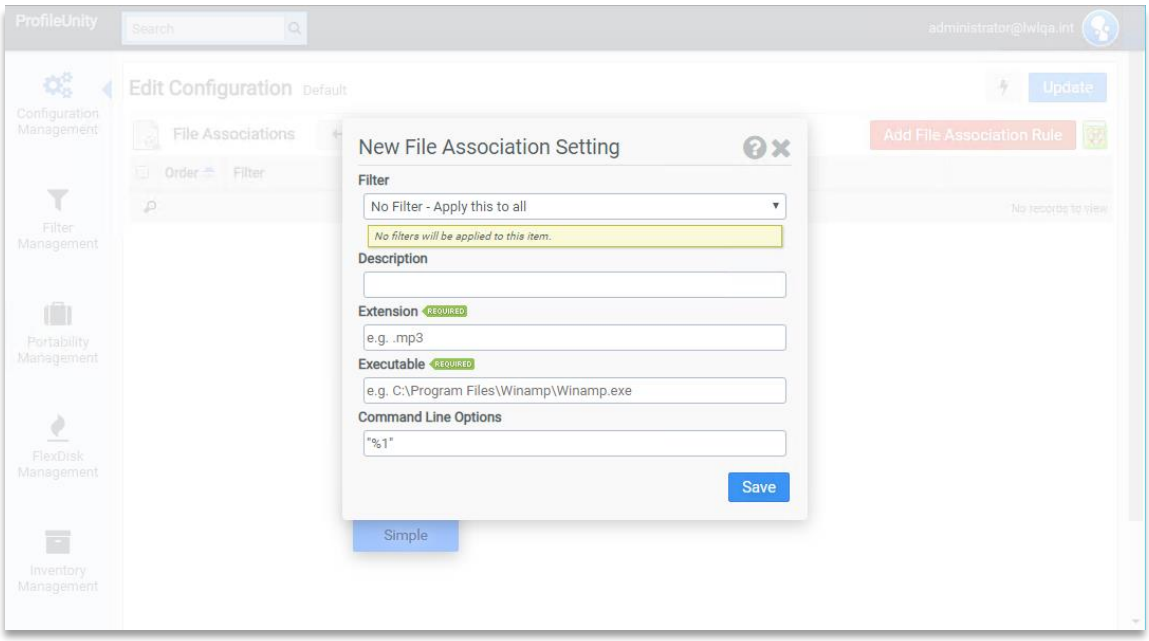
Example 2:

Filter: No Filter – Apply this to all
Action: Save/Restore
Ruleset: Application Data
Apply Registry Rules: Enabled
Isolate Subkeys: False
Apply Filesystem Rules: Disabled
Isolate Subdirectories: False
Save Path: \\ATLFS02\@userid\Citrix_Profile
Restore Once Tracking Path: blank
UID: wjdemgosiitd1vs

Example 2 saves the settings defined by the `Application Data ruleset` at logoff and restores them at logon. Only registry rules defined in the ruleset will be applied. The settings will be stored and retrieved from a file named `wjdemgosiitd1vs.7z` located in the `\\ATLFS02\@userid\Citrix_Profile` folder. This example leverages the macro `@USERID`. When ProfileUnity executes, the `@USERID` macro is expanded into the username.

File Associations

The File Associations module is used to create a mapping between a particular file extension and an associated executable, allowing you to control which default application should be started when using a certain file type. Optional command line options to use when running the executable may be specified as well. Admins can control which users are assigned which File Association rules by using Filters.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Extension:
Enter the file extension for which you are creating a mapping.

Executable:
Enter the executable you wish to associate with the extension. Please note that some system applications like Notepad do not require a full path, just "notepad.exe", while third party applications will require the full path: "C:\Program Files\yourapp.exe".

Command Line Options:
Optionally enter any command line options that the executable should use when running.

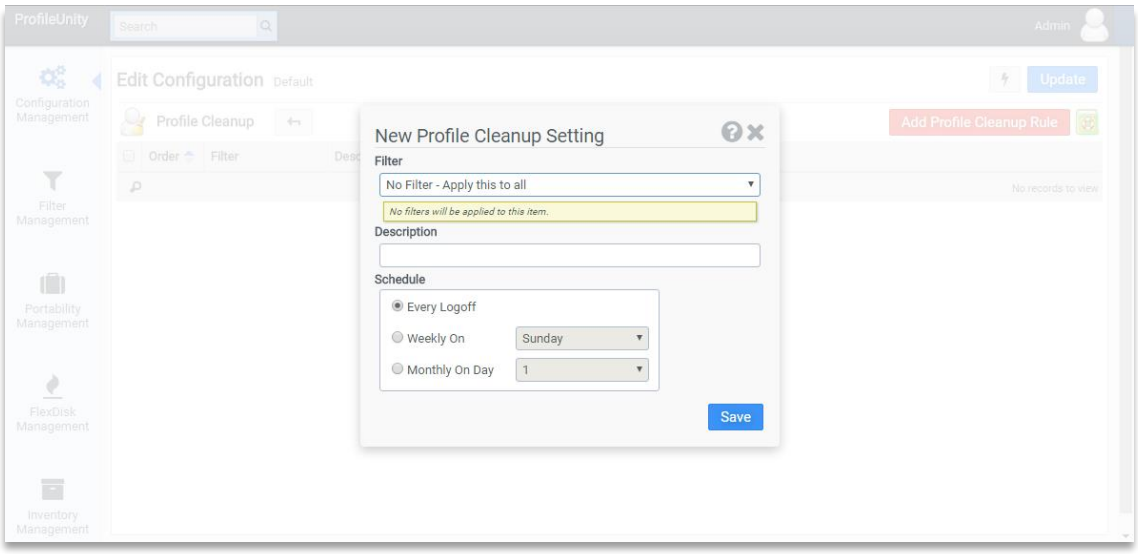
Example:

Filter: No Filter - Apply this to all
Extension: .mp3
Executable: C:\Program Files\Winamp\winamp.exe

The above example will associate .mp3 files with the Winamp application.

Profile Cleanup

The Profile Cleanup module is used to clean up a user's profile contents. The Profile Cleanup rules are evaluated at user login. If the cleanup schedule for the user matches the desktop's day of the week or day of the month, then a flag will be set to remove the user's profile from that desktop at the user's next logoff. Profile Cleanup applies to Windows 7 and higher or Windows Server 2008 R2 machines and higher only.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Schedule:
Select when this cleanup operation should be performed. Choose from:

- **Every Logoff**
- **Weekly and** select a particular day of the week
- **Monthly** and select a particular day of the month

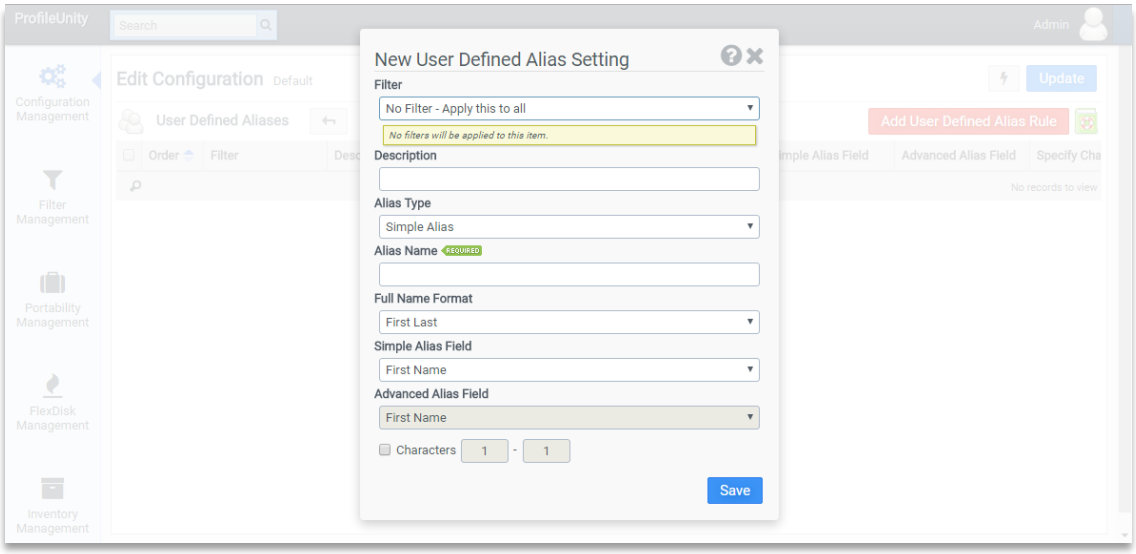
Example:

Filter: No Filter - Apply this to all
Schedule: Every Logoff

The above example will cause a user's profile to be removed from the system upon every logoff for all users.

User Defined Aliases

User Defined Aliases are used to retrieve information from your domain controllers about the current user's account. The values retrieved are available for use throughout your ProfileUnity configuration.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Alias Type:
A **Simple Alias** queries the domain controller for the Full Name field. An **Advanced Alias** queries an Active Directory domain controller for multiple attributes assigned to a user account. The **Advanced Alias** option is more powerful and retrieves more information than the **Simple Alias** option. However, the **Advanced Alias** option cannot be used in every environment.

Alias Name:
Enter a descriptive name for your alias. This name can be used throughout your configuration and will be dynamically substituted with the value retrieved from the domain controller. An **Alias Name** is marked for substitution by prepending it with two dollar signs. For example, if you choose `fname` as your **Alias Name**, you need to use `$$fname` anywhere you want the substitution to occur.

Full Name Format:
Select the format of the Full Name field. This option applies to **Simple Aliases** only. Choose from:

- **First Last**
- **First Middle Last**
- **Last, First**
- **Last, First Middle**

Simple Alias Field:
The **Simple Alias Fields** are derived from the Full Name field. Select the field containing the values you want assigned to your **Alias Name**. This option applies to **Simple Aliases** only. Choose from:

- **First Name**
- **Middle Name**
- **Last Name**

Advanced Alias Field:
The **Advanced Alias Field** corresponds to the fields listed in Active Directory. Select the field containing the value you want assigned to your **Alias Name**. This option applies to **Advanced Aliases** only.

Characters:
This allows you to use only part of the value retrieved from the domain controller. If you enable this option, you will need to choose a starting character and an ending character. For example, entering one (1) for the starting character and eight (8) for the ending character would retrieve the first eight (8) characters.

Example 1:

Filter: No Filter – Apply this to all
Alias Type: Advanced Alias
Alias Name: email
Advanced Alias Field: E-mail

Example 1 creates an alias named `email`. This alias will contain the user's E-mail address retrieved from Active Directory. Anywhere you use `$$email` in your configuration, the user's E-mail address will be substituted.

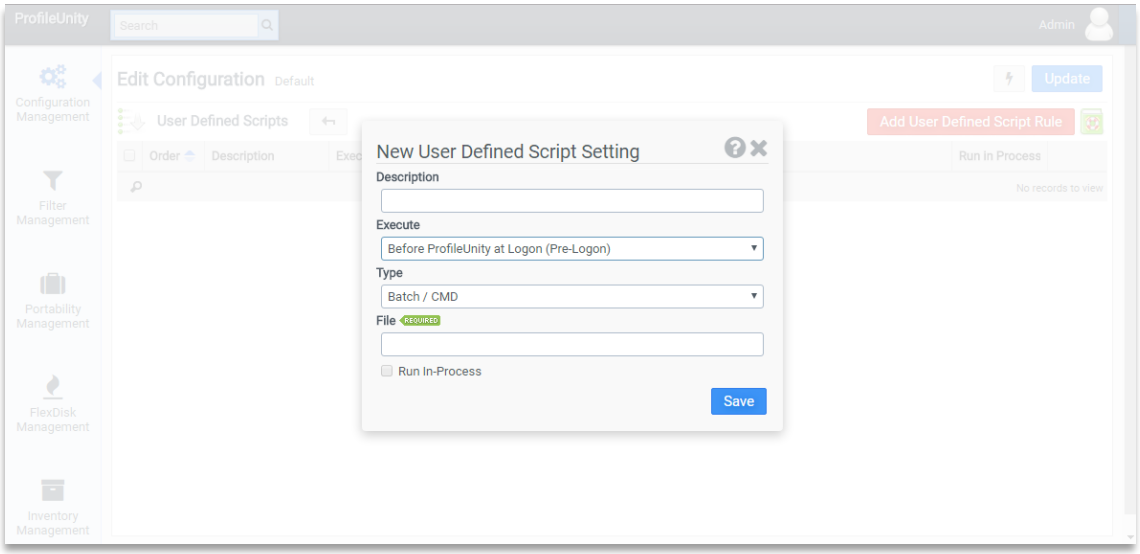
Example 2:

Filter: No Filter – Apply this to all
Alias Type: Simple Alias
Alias Name: lname_initial
Full Name Format: First Middle Last
Simple Alias Field: Last Name
Characters: 1-1

Example 2 creates an alias named `lname_initial`. This alias will contain the first character of the user's Last Name. The user's Last Name is derived from the Full Name Format option and the Full Name value retrieved from the authenticating domain controller. Anywhere you use `$$lname_initial` in your configuration, the first character of the user's Last Name will be substituted.

User Defined Scripts

Administrators wishing to utilize a custom written script will add it to their configuration here. ProfileUnity offers the option to include custom scripts both before ProfileUnity (Pre-Execution) and after ProfileUnity (Post-Execution).



Description:
Enter a description for this rule.

Execute:
Select when your custom script should execute:

- **Before ProfileUnity at logon (Pre-logon)**
- **After ProfileUnity at logon (Post-logon)**
- **Before ProfileUnity at logoff (Pre-logoff)**
- **After ProfileUnity at logoff (Post-logoff)**

Type:
Select the type of script you are including:

- **Batch/CMD**
- **KiXtart**
- **Windows Script Host**
- **Executable**
- **PowerShell**

File:
Enter the full path to your script. Surrounding quotes are automatically added to the file.

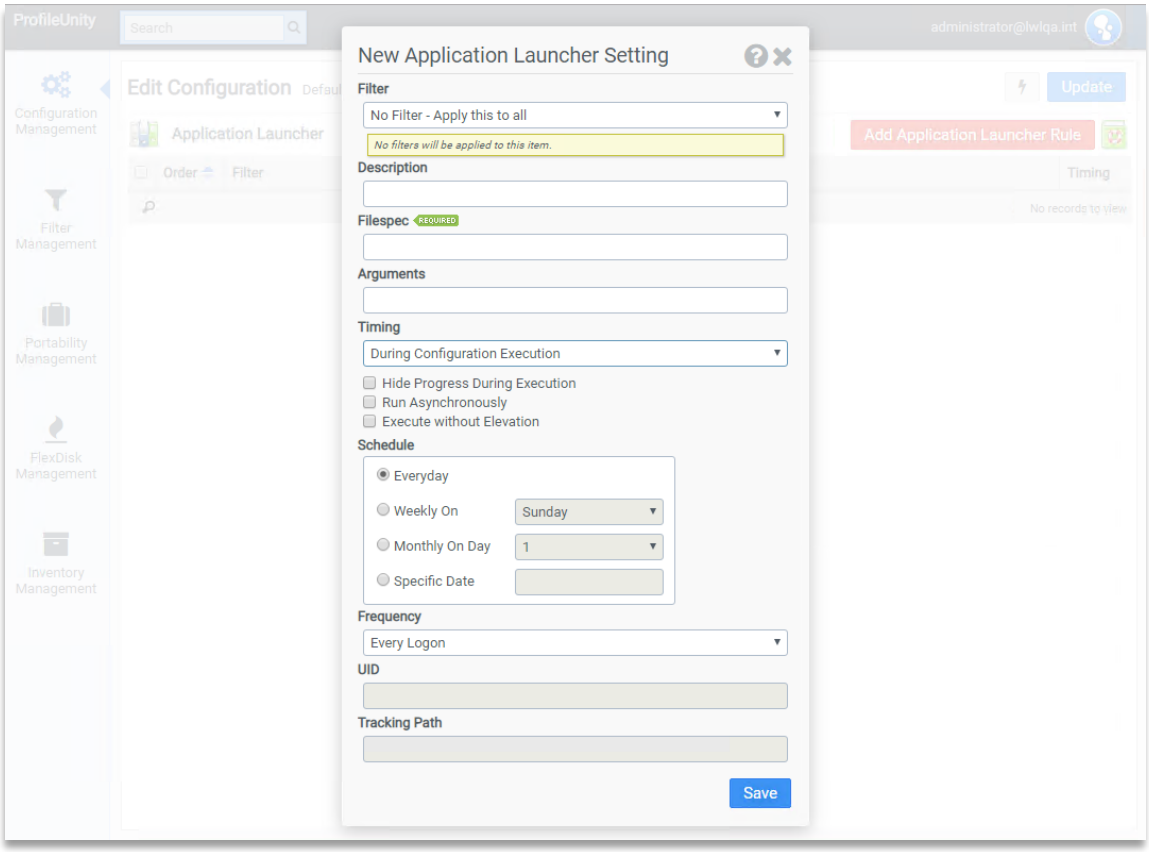
NOTE: The recommended location to store your custom scripts is in the NETLOGON share on your domain controller. Custom scripts stored in this location will take advantage of replication and can be included as follows:
@lserver\netlogon\[your_script].kix.

Run In-Process:
Selecting this option will merge your custom KiXtart script into ProfileUnity. This option should only be selected if you want all or parts of your script exposed to ProfileUnity. For example, if you are utilizing a Custom Function (deprecated feature) as part of a filter, this option must be selected.

NOTE: Custom scripts are not checked for proper syntax or function. You are encouraged to create and test your custom scripts prior to inclusion with the User Defined Scripts module.

Application Launcher

This module allows you to launch applications on your client machines during or after ProfileUnity execution.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Filespec:
Enter the full path to the executable. Surrounding quotes are automatically added to the filespec.

Arguments:
Enter any arguments required by the executable. **Arguments** are not automatically quoted.

Timing:
Choose whether your application should launch during or after ProfileUnity execution.

Hide Progress During Execution:
Selecting this option will hide the ProfileUnity progress display while your application executes. This option is not available if you enable **Run Asynchronously**.

Run Asynchronously:
By default, ProfileUnity waits for your application to terminate prior to continuing. When this option is selected, ProfileUnity will not wait for your application to terminate.

Execute without Elevation:
Select this option to run the application without privilege elevation.

Schedule:
Choose the schedule your application should execute on:

- **Everyday**
- **Weekly and** select a particular day of the week
- **Monthly** and select a particular day of the month
- **Specific Date** and select the date

Frequency:
Choose the frequency your application should execute on:

- **Every Logon**
- **Once Per Day (Computer)**
- **Once Per Day (User)**
- **One Time (Computer)**
- **One Time (User)**

UID:
The **UID** is a unique identifier that is automatically generated by ProfileUnity. Normally, the **UID** does not need to be changed. Changing the **UID** will reset the execution history associated with the application. For example, if you chose a frequency of One Time (User) and you want the application to run a second time, changing the **UID** will cause the application to execute again. This option does not apply if you choose a **Frequency** of **Every Logon**.

Tracking Path:
The **Tracking Path** designates the UNC name of the network share ProfileUnity will use to store execution history. When you create the share, you will need to grant write permission to all users using ProfileUnity. If this share is unavailable or does not have write permission, ProfileUnity will not execute the application. This option does not apply if you choose a **Frequency** of **Every Logon**.

NOTE: Inside the share ProfileUnity will create a folder named after the UID of the application. Therefore, it is safe to use the same share as the Tracking Path for multiple applications. If you wish to reset the execution history for an application, you can delete the folder associated with the application's UID.

Example:

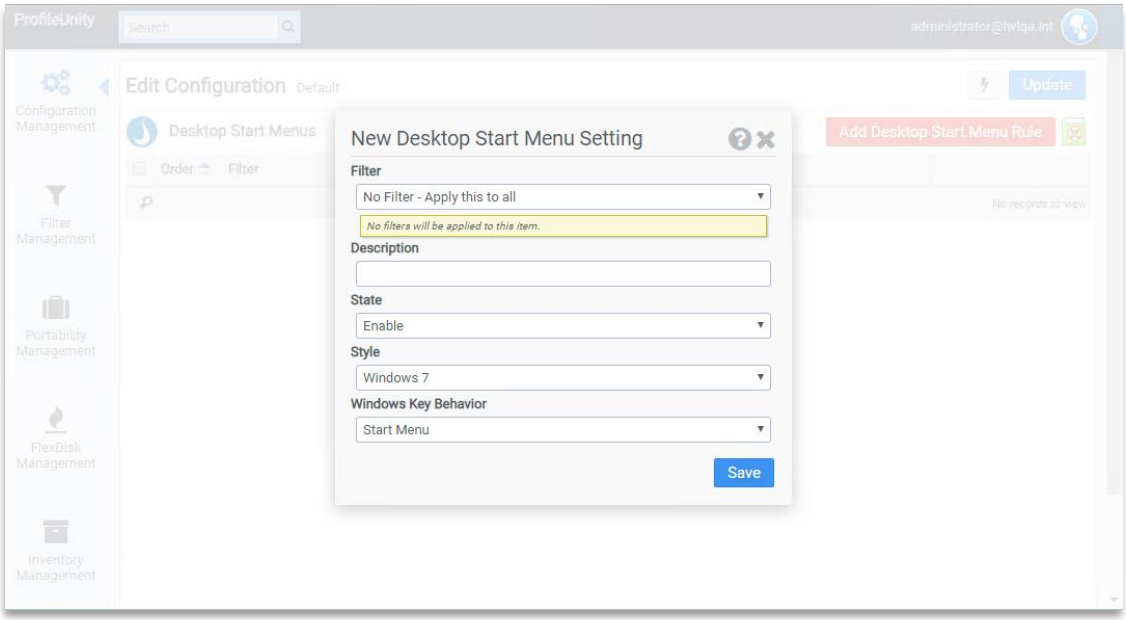
Filter: No Filter – Apply this to all
Filespec: C:\Program Files\Adobe\Acrobat 6.0\Acrobat\Acrobat.exe
Arguments: "\\ATLFS01\public\Employee Handbook.pdf"
Timing: After Configuration Execution
Run Asynchronously: Enabled
Schedule: Everyday
Frequency: One Time (User)
UID: gtnj8nvrp4oe38m
Tracking Path: \\ATLFS02\apptrack$$

The above example will launch Adobe Acrobat and open the file `Employee Handbook.pdf` one time for each user. Since Run Asynchronously has been enabled, ProfileUnity will not wait for the user to close the Acrobat reader. The Arguments passed to the Filespec have been surrounded with quotes as `Employee Handbook.pdf` contains a space.

NOTE: The share name used for the Tracking Path is \\ATLFS02\apptrack$. Since $ is a reserved character, it has been escaped with $$. Please Appendix C for assistance using reserved characters.

Desktop Start Menus

The Desktop Start Menus module allows a user to add a classic start menu to Windows 8.1 and Windows Server 2012 R2 desktops. By default, this module is hidden. To use this configuration module, go to **Administration > Settings > Miscellaneous** and uncheck the "Hide Desktop Start Menu Module".

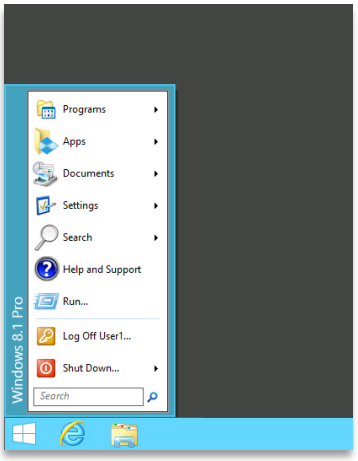


Filter:
Select the name of the filter you want assigned to this configuration element.

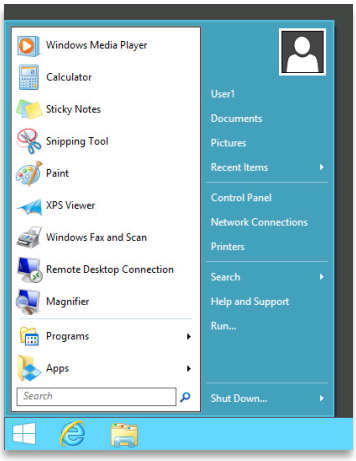
Description:
Enter a description for this rule.

State:
Choose whether to **Enable** or **Disable** the start menu on the desktop.

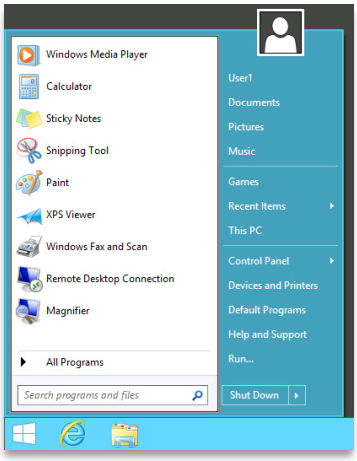
Style:
Choose which style of start menu to use: **Windows XP, Windows XP Two Columns, or Windows 7.**



Windows XP



Windows XP Two Columns



Windows 7

Windows Key Behavior:
Indicates how the user's Windows key should behave when pressed. Choose to bring up the **Start Menu** or the **Windows 8 Metro** Start Screen.

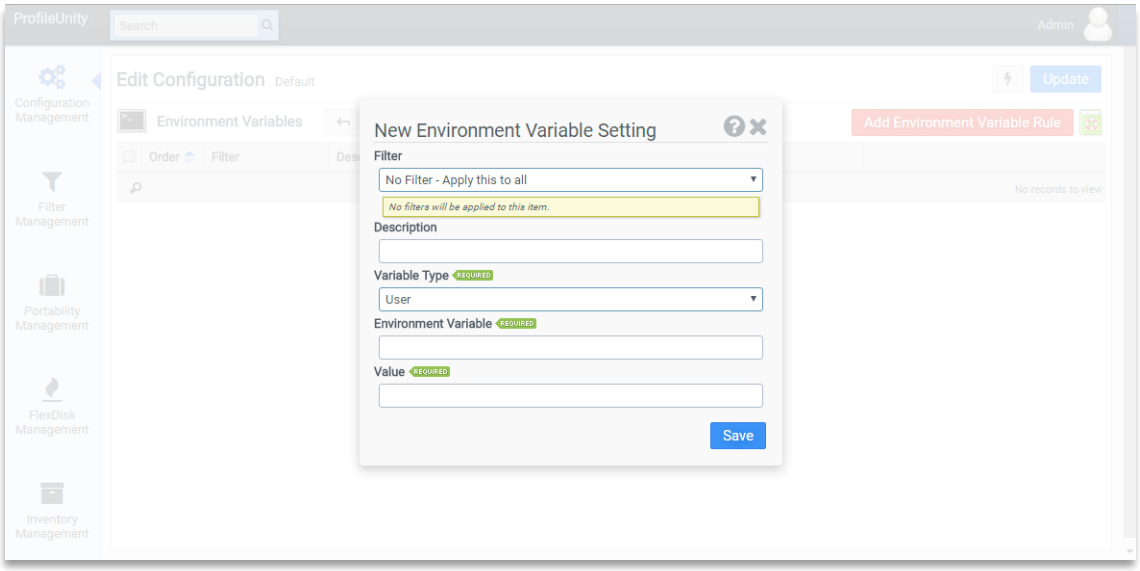
Example:

Filter: No Filter – Apply this to all
State: Enable
Style: Windows 7
Windows Key Behavior: Start Menu

The above example will enable the use of a desktop start menu similar to Microsoft's Windows 7 Start Menu.

Environment Variables

The Environment Variables module sets and clears environment variables. Environment variables are modified in the environment of the current user.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Variable Type:
Select whether this is a **User** or **System** variable.

Environment Variable:
Enter the name of the environment variable you want to modify. The name should not be enclosed in percent signs.

Value:
Enter the value you want assigned to the environment variable. Leaving this value blank will delete the environment variable.

Example:

Filter: No Filter – Apply this to all
Variable Type: User
Environment Variable: MYDIR
Value: C:\MyDir

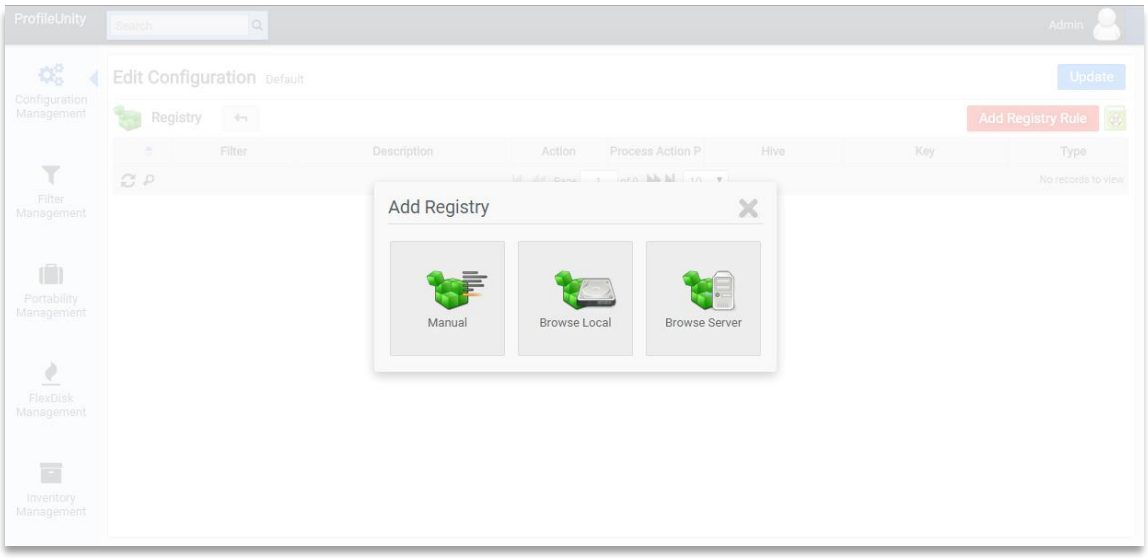
The above example assigns the value `C:\MyDir` to the environment variable `MYDIR`.

Registry

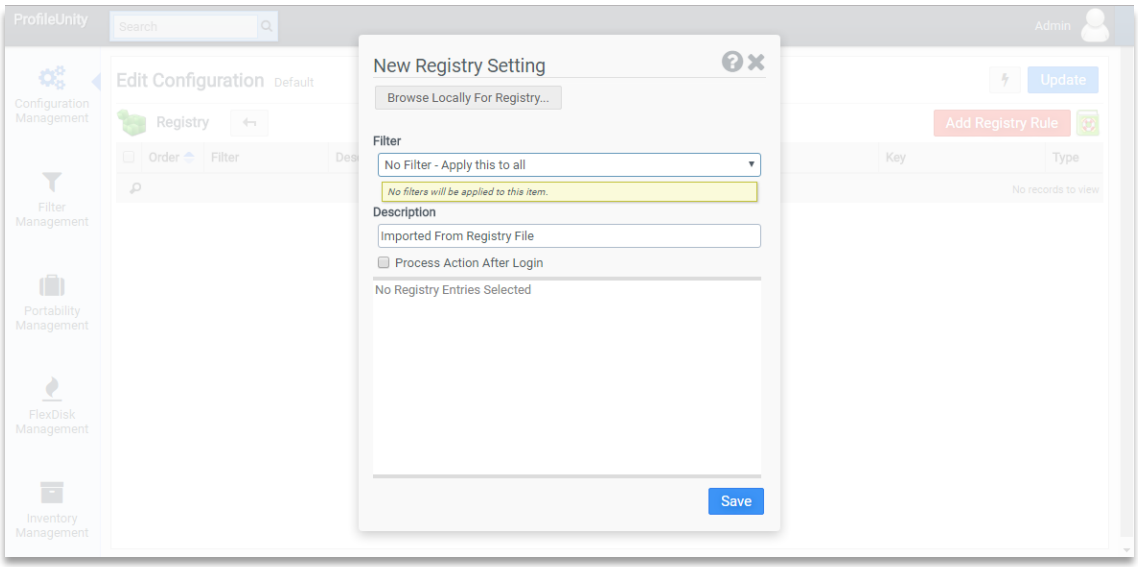
This module allows you to make changes to the registry on your client machines.

Choose whether to add registry settings manually, browse local settings, or browse server settings to select registry settings.

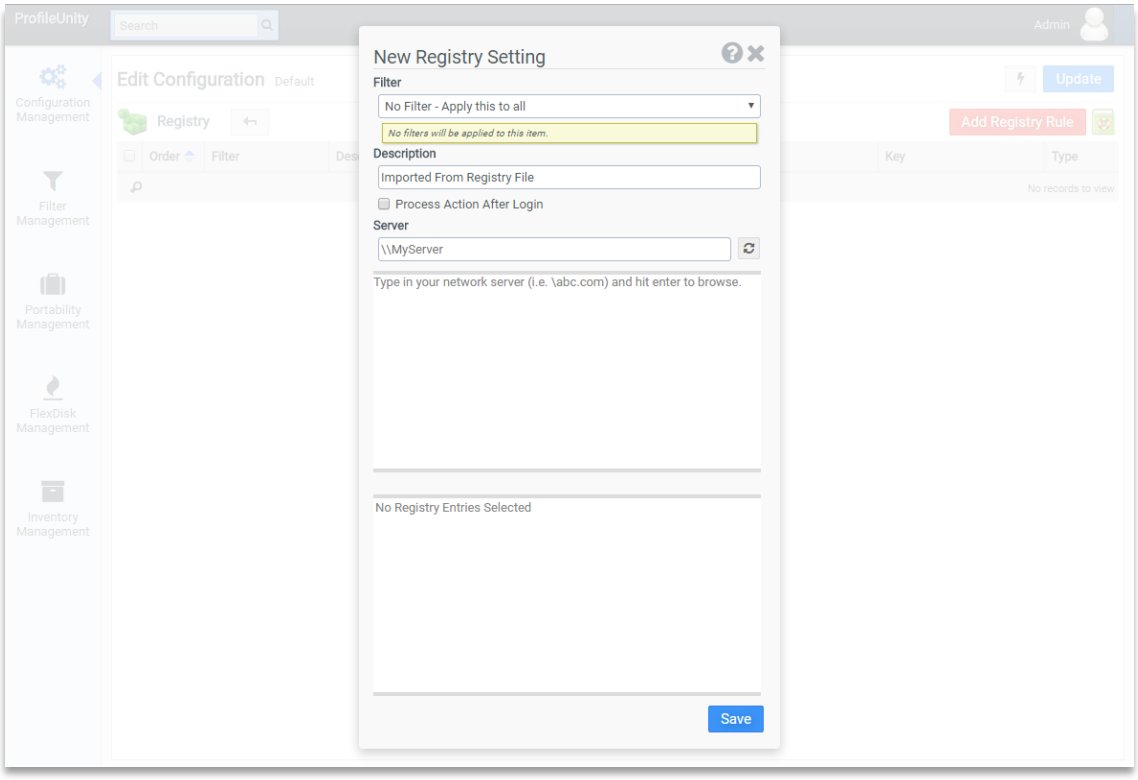
Note: This configuration module has moved. Formerly, the Registry module followed the RDP Client module. Starting with 6.5, the Registry module now follows the Environment Variable module. Since most module settings are processed in the same order as they appear in the Configuration Management area, it is possible for behavioral differences to exist between older and newer configurations created with different versions of the ProfileUnity Management Console.



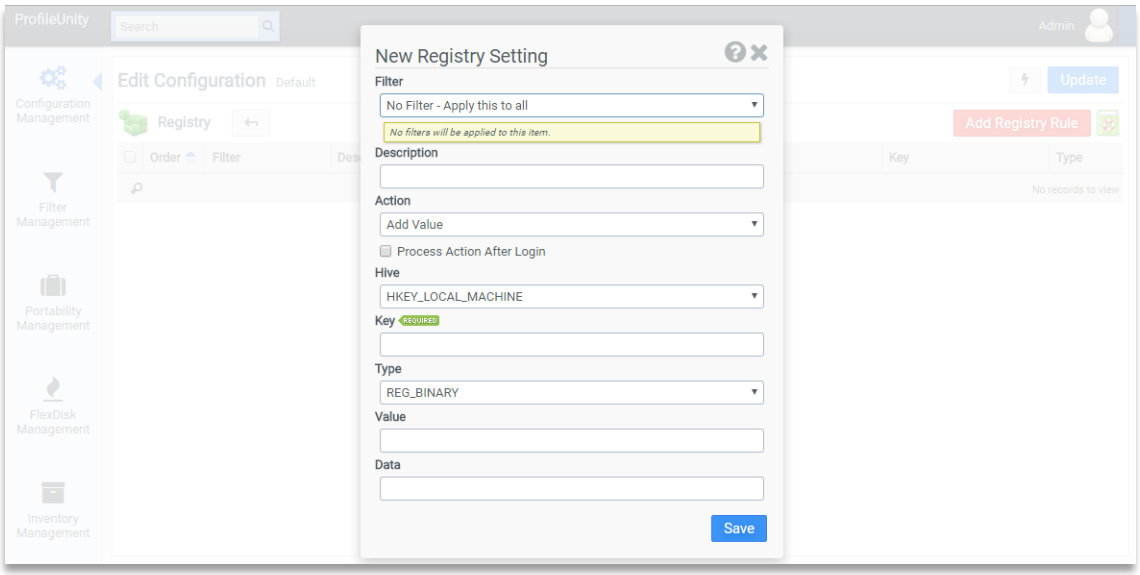
Select **Browse Local** and then click the "**Browse Locally for Registry…**" button to search locally for registry settings.



Select **Browse Server** and then type your server name and press **Enter** to search a server for registry settings.



Select **Manual** to setup registry settings manually.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Action:
Select one of the following actions:

- **Add Value** – Modifies or creates a value and assigns data to it.
- **Delete Value** – Deletes a value and associated data.
- **Add Key** – Adds the specified key.
- **Delete Key** – Deletes the specified key.

Process Action Post Login:
Process the action after a user logs in. Moving select rules to process after login can speed up login times.

Hive:
Select the registry hive containing the key or value you want to modify. Choose from:

- **HKEY_LOCAL_MACHINE**
- **HKEY_USERS**
- **HKEY_CURRENT_CONFIG**
- **HKEY_CLASSES_ROOT**
- **HKEY_CURRENT_USER**

Key:
Enter the subkey of the Hive you want to modify.

Type:
Select the **Type** of the Value you want to modify. This option is only available if **Write Value** is selected as the **Action**. Choose from:

- **REG_BINARY**
- **REG_DWORD**
- **REG_EXPAND_SZ**
- **REG_MULTI_SZ**
- **REG_SZ**
- **REG_QWORD**

Value:
Enter the value you want to modify. This option is only available if **Write Value** or **Delete Value** is selected as the **Action**.

Data:
Enter the data you want assigned to the value. This option is only available if **Write Value** is selected as the **Action**.

Example 1:

Filter: No Filter – Apply this to all
Action: Add Key
Hive: HKEY_CURRENT_USER
Key: Software\MyCompany\Test

Example 1 adds the key `Software\MyCompany\Test` to the `HKEY_CURRENT_USER` hive.

NOTE: The Add Key Action creates missing subkeys automatically. In the above example, the `MyCompany` subkey will be created if it does not already exist.

Example 2:

Filter: No Filter – Apply this to all
Action: Write Value
Hive: HKEY_CURRENT_USER
Key: Software\Microsoft\Windows\CurrentVersion\Internet Settings\Cache
Type: REG_DWORD
Value: Persistent
Data: 0

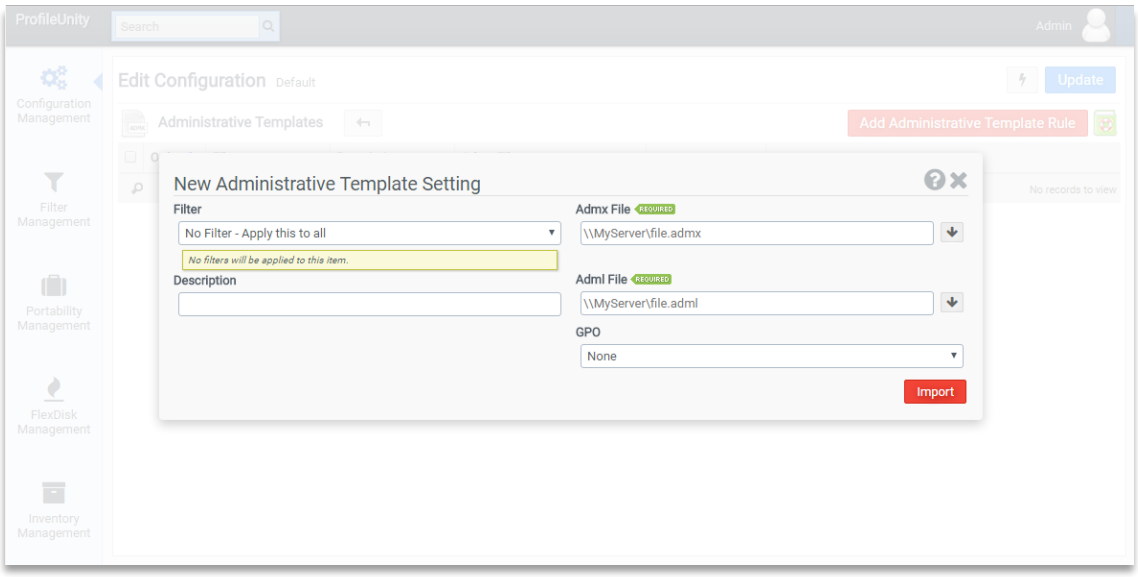
Example 2 demonstrates using the Registry module to update Internet Explorer settings. Setting the Persistent value to 0 instructs Internet Explorer to empty the temporary Internet files folder when the browser is closed.

NOTE: The Write Value Action creates missing subkeys automatically. In the above example, the subkey `Software\Microsoft\Windows\CurrentVersion\Internet Settings\Cache` will be created if it does not already exist.

Administrative Templates

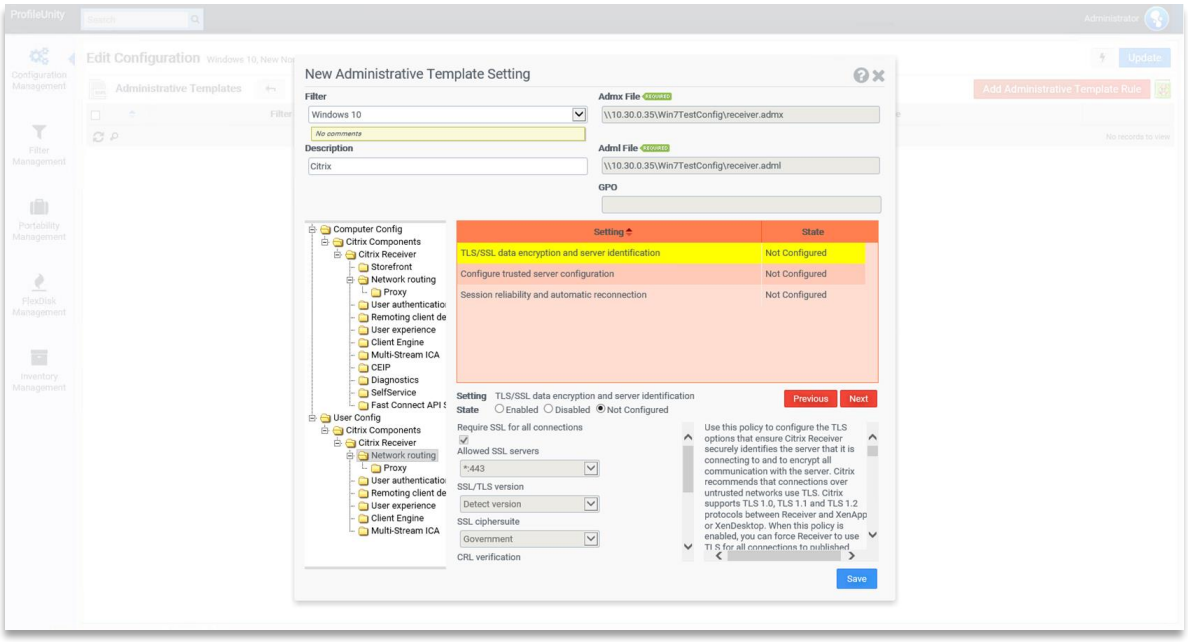
The Administrative Templates module allows administrators to import and configure Windows Active Directory ADMX/ADML files containing registry-based policy settings, providing web-based Group Policy Management. ProfileUnity can process the environment's GPOs once they have been imported into the Configuration Management Console and disabled in Active Directory.

Enter the server name to browse for the AMDX and AMDL file to import.



Once a pair of ADMX and ADML files are imported, determine which settings to enable, disable, or leave not configured. For enabled settings, input custom values into the provided controls (if any). Here is more information on what the various states mean:

- **Not Configured** – This is the default state for all policy settings. This means that this setting will not affect the users or computers in any way. No registry changes will be made.
- **Enabled** – This activates the policy setting. Registry values may be added, changed or even deleted.
- **Disabled** – The policy setting will delete associated values in the Registry.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

ADMX File:
Select and ADMX file. ADMX files provide structural information about Group Policy configurations.

ADML File:
Select and ADML file. ADML files provide language specific string translations to make Group Policies readable in different languages.

GPO:
Optionally select a GPO. If selected, the administrative template will be populated with values from that GPO that are part of the input ADMX/ADML.

Example:

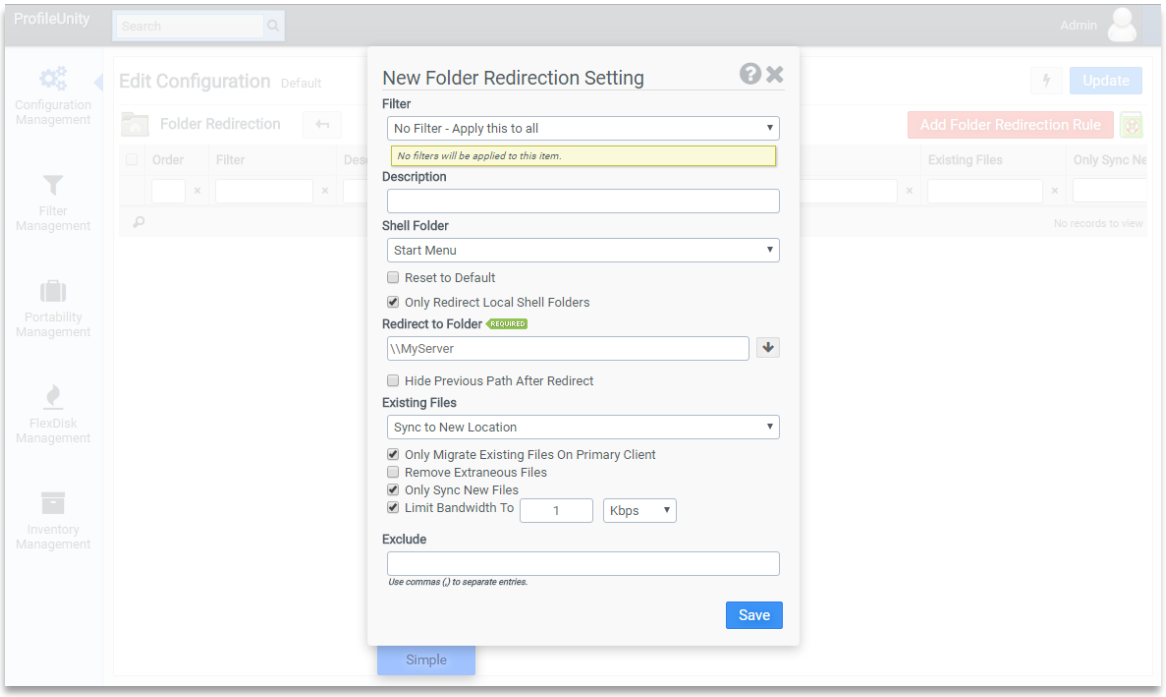
Filter: No Filter – Apply this to all
Description: My first admin template
ADMX File: c:\windows\policydefinitions\mswebsitesample.admx
ADML File: c:\windows\policydefinitions\en-us\mswebsitesample.adml
GPO: none

The above example imports the ADMX and ADML files mswebsitesample into the configuration.

Folder Redirection

Shell folders are special folders that Windows uses to indicate the default location for many types of settings and data. This module allows you to redirect shell folders to an alternate location, and optionally move or copy existing data. A common use for this module is redirecting "My Documents" to a network share.

All options for the Advanced settings are shown below.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Shell Folder:
Select the Shell Folder you want to redirect.

Reset to Default:
Enabling this option will reset the selected Shell Folder to its default location.

Only Redirect Local Shell Folders:
When this option is checked and the shell folder is a UNC path or Mapped Drive, the folder will be skipped and not synced. If the shell folder is on any drive letter it will be supported for sync.

Redirect to Folder:
Enter the path you want the Shell Folder redirected to. This option is not available if **Reset to Default** has been selected.

Hide Previous Location After Redirection:
Enabling this option causes ProfileUnity to set the hidden attribute on and rename the previous shell folder location after redirection.

Existing Files:
Files and folders that exist in the current Shell Folder location can optionally be copied, moved, or synchronized to the new location. Depending on the amount of data that currently exists, enabling copy or move can cause a delay during logon while the files and folders are copied or moved to the new location. Alternatively, the synchronize option will transfer the existing files and folders in the background without impacting the user's logon time. The synchronize option firsts performs a background synchronization of the user's existing files and folders. At the next logon, after the initial background synchronization has completed, foreground synchronization takes place to synchronize any missing files or folders. Once the foreground synchronization has completed, the folder is then redirected. The user does have to wait for the foreground synchronization to complete during logon. However, the foreground synchronization is usually quick whereas it is only moving a few missing files and folders. The synchronize option is the recommended method for migrating existing files and folders. This option is not available if **Reset to Default** has been selected.

Only Migrate Existing Files on Primary Client:
Enabling this option causes ProfileUnity to only migrate existing files on the user's Primary Client. The user's Primary Client is the first computer a user logs on to after a folder redirection is configured.

Remove Extraneous Files:
Enabling this option will delete files and folders that exist in the new location and do not exist in the current location during synchronization. This option is only available if **Sync to New Location** or **Sync Only – Don't Redirect** is chosen for **Existing Files**.

NOTE: This option can cause data loss if not used with care. It is recommended that this option be left disabled unless you have a specific need to enable it.

Only Sync New Files:
Enabling this option will allow conflict resolution when syncing from many computers or when trying to start the sync process over again to only send the newer files. This option is only available if **Sync to New Location** or **Sync Only – Don't Redirect** is chosen for **Existing Files**.

Limit Bandwidth to:
Enabling this option will limit the amount of bandwidth the background synchronization process consumes. This option is only available if **Sync to New Location** or **Sync Only – Don't Redirect** is chosen for **Existing Files**.

Exclude:
Paths entered will be excluded from the background synchronization process. Use commas to separate multiple paths to be excluded. This option is only available if **Sync to New Location** is chosen for **Existing Files**. The following matching rules apply:

- a leading '\' anchors the match to the full path only (otherwise partial matches occur)
- a '*' matches any path component except a backslash
- a '?' matches any character except a backslash
- a trailing '\' limits the match to directories and not files

Example:

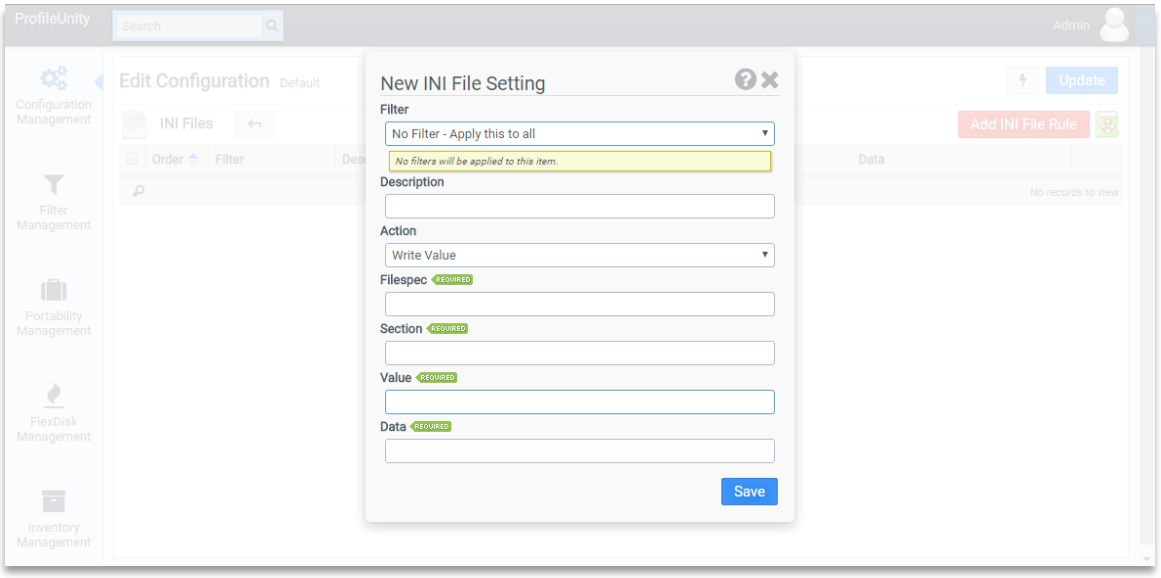
Filter: No Filter – Apply this to all
Shell Folder: Personal (My Documents)
Reset to Default: Disabled
Redirect to Folder: H:\My Documents
Existing Files: Copy to New Location

The above example redirects the `My Documents` Shell Folder to `H:\My Documents`. All existing files and folders are copied to the new location.

NOTE: This example assumes the H: drive is mapped to a network share. It is also possible to use a UNC path for the Redirect to Folder.

INI Files

Many programs rely on INI files for their configuration. This module enables ProfileUnity to manipulate INI files during user logon.



Here is the structure of an INI file.

```
[Section]
Value=Data
```

INI FILE STRUCTURE

Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Action:
Select one of the following actions:

- **Write Value** – Modifies or creates a value and assigns data to it.
- **Delete Value** – Deletes a value and associated data.
- **Delete Section** – Deletes an entire section.

Filespec:
Enter the full path to the INI file. If this parameter does not include a full path, ProfileUnity searches for the file in the Windows directory.

Section:
Enter the name of the section to be modified in the INI file. If the section does not exist, it is created. The section name is not case-sensitive, and can contain any combination of uppercase and lowercase letters.

Value:
Enter the name of the value to be modified in the INI file. If the value does not exist, it is created. This option is not available if **Delete Section** is selected as the **Action**.

Data:
Enter the data you want assigned to the value. This option is only available if **Write Value** is selected as the **Action**.

Example:

Filter: No Filter – Apply this to all
Action: Write Value
Filespec: %APPDATA%\ICAClient\pn.ini
Section: ENTRIGUE
Value: HttpBrowserAddress
Data: ica.entrigue.net

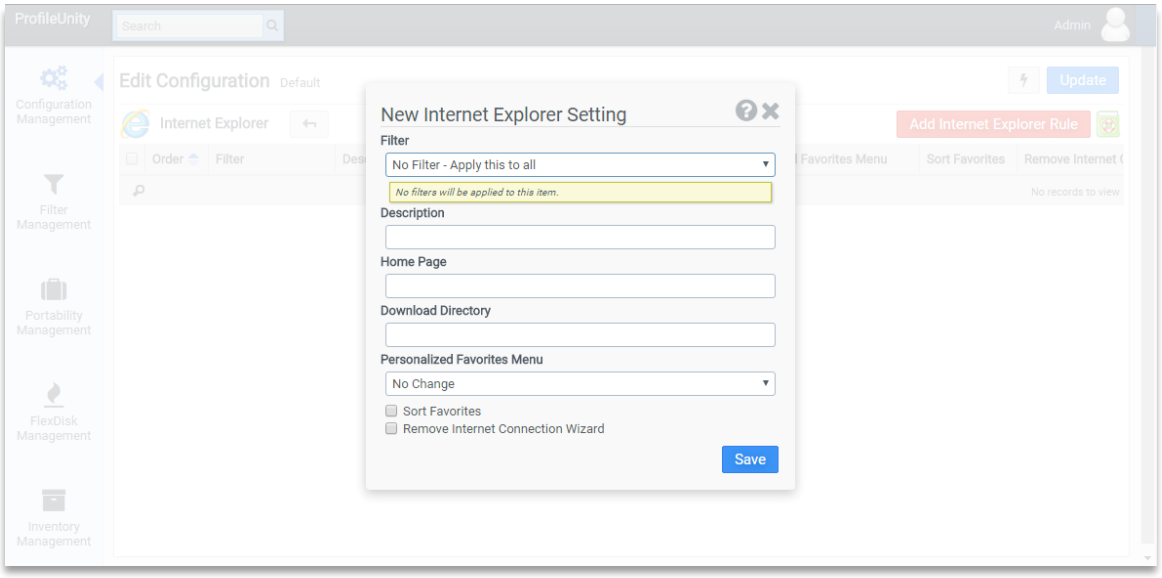
The above example demonstrates using the INI Files module to update Citrix Program Neighborhood settings. The `HttpBrowserAddress` for the application set named `ENTRIGUE` will be set to `ica.entrigue.net`. The `pn.ini` file will have the changes made to it shown here.

```
[ENTRIGUE]
HttpBrowserAddress=ica.entrigue.net
```

PN.INI CHANGES

Internet Explorer

This module is used to configure Internet Explorer features such as Start Page, Download Directory, and Favorites. This module supports Internet Explorer versions 4.0 and higher.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Start Page:
Enter the URL of the page you want Internet Explorer to display on startup. If you want Internet Explorer to start with a blank page, enter `about:blank` for the **Start Page**.

Download Directory:
The **Download Directory** is the location presented to the user when asked to save a file downloaded with Internet Explorer. Enter the full path to the folder you want to use.

Personalized Favorites Menu:
The favorites menu is automatically updated to keep the favorites you use most often at the top of the menu where they are usually more accessible. This option allows you to control the Personalized Favorites Menu setting used by your users. Available choices are: **Enable**, **Disable**, and **No Change**.

Sort Favorites by Name:
Selecting this option will sort the Internet Explorer Favorites menu alphabetically.

Remove Internet Connection Wizard:
Selecting this option will disable the Internet Connection Wizard and remove the Internet Connection Wizard desktop shortcut.

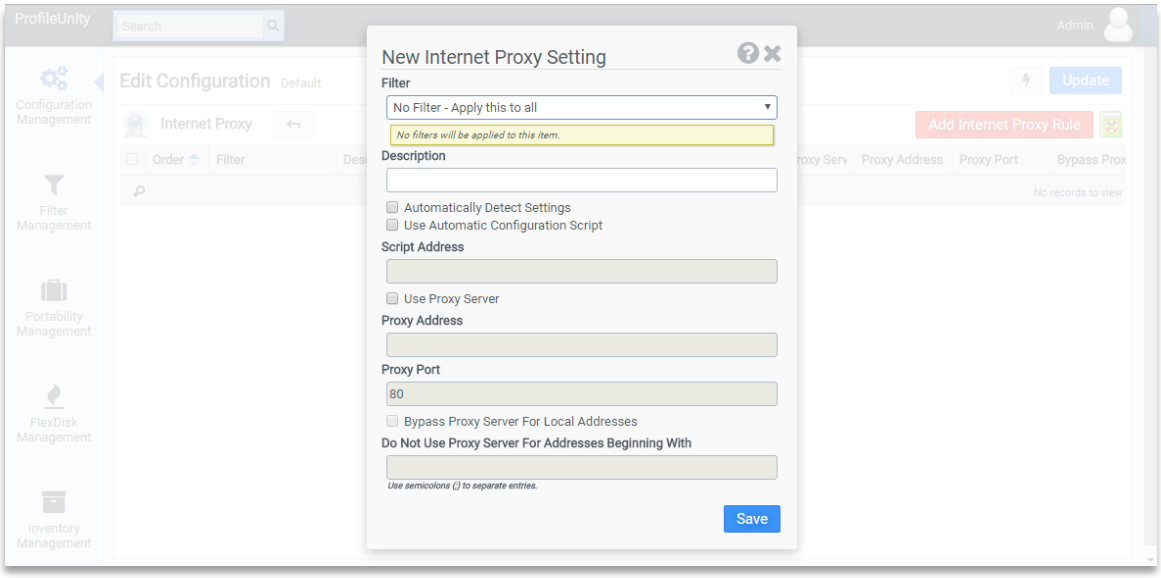
Example:

Filter: No Filter – Apply this to all
Start Page: http://www.entrigue.net
Download Directory: C:\Temp
Personalized Favorites Menu: Disable
Sort Favorites by Name: Not Selected
Remove Internet Connection Wizard: Selected

The above example sets the Internet Explorer start page to `http://www.entrigue.net`, sets the download directory to `C:\Temp`, disables Personalized Favorites, and removes the Internet Connection Wizard.

Internet Proxy

This module is used to adjust Internet Explorer's proxy settings to match your network configuration. This module supports Internet Explorer versions 9.0 and higher and Microsoft Edge.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Automatically Detect Settings:
Selecting this option enables the browser's automatic proxy detection feature. This option is only available on Internet Explorer versions 8.0 and higher and on Microsoft Edge.

Use Automatic Configuration Script:
Select this option if you want to configure browser proxy settings with a configuration script. You will need to enter the complete path to the configuration script in the **Script Address** field.

Script Address:
Enter the complete path to the configuration script you are using. This option is only available if **Use Automatic Configuration Script** is selected.

Use a Proxy Server:
Selecting this option instructs the browser to connect to the Internet by using the proxy server settings you specify in the **Proxy Address** and **Proxy Port** fields.

NOTE: You can configure the browser to use different proxy servers for different protocols by leaving the Port field blank and using the following format for the Address field: `protocol=address:port`. The available protocols are `ftp`, `gopher`, `http`, `https`, and `socks`. You will need to separate multiple entries with semicolons. For example:

`http=proxy1.company.com:8000;ftp=proxy2.company.com:3128.`

Proxy Address:
Enter the proxy server address. This option is only available if **Use a Proxy Server** has been selected.

Proxy Port:
Enter the proxy server port. This option is only available if **Use a Proxy Server** has been selected.

Bypass Proxy Server for Local Addresses:
This option instructs the browser to bypass the configured proxy settings for local addresses. This option is only available if **Use a Proxy Server** has been selected.

Do Not Use Proxy Server for Addresses Beginning With:
Enter addresses that the browser should access without using the configured proxy settings. This option is only available if **Use a Proxy Server** has been selected.

NOTE: You can use wildcards to match addresses. For example: `*.xyz.com, 192.168.*.`

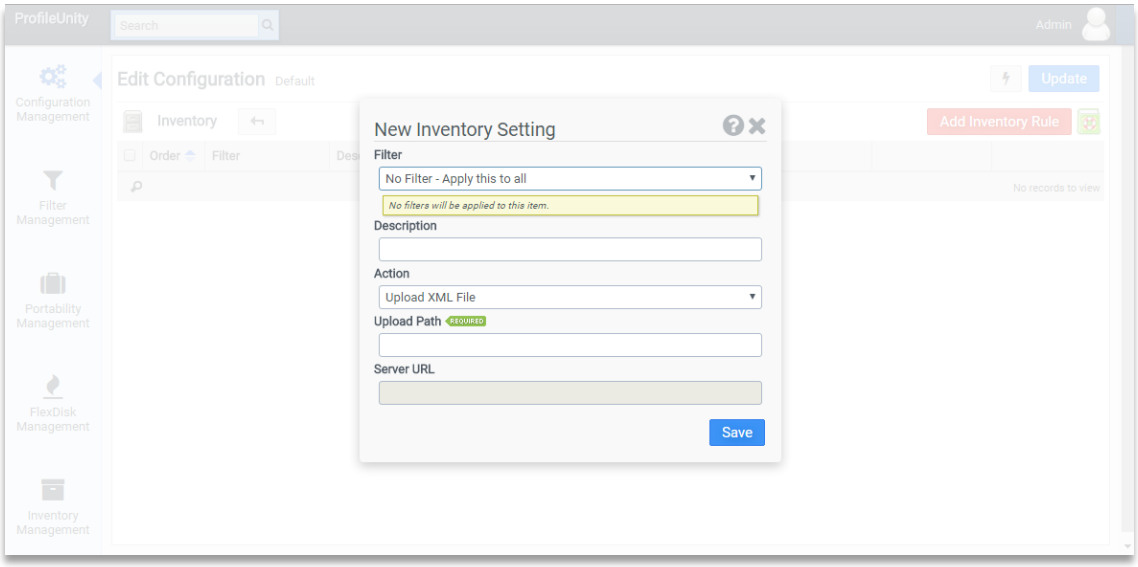
Example:

Filter: No Filter – Apply this to all
Automatically Detect Settings: Not Selected
Use Automatic Configuration Script: Not Selected
Use a Proxy Server: Selected
Address: proxy.liquidwarelabs.com
Port: 8000
Bypass Proxy Server for Local Addresses: Selected
*Do Not Use Proxy Server for Addresses Beginning With: *.liquidwarelabs.com*

The above example configures Internet Explorer to use proxy server `proxy.liquidwarelabs.com` on port `8000`. The proxy server will not be used for local addresses or for addresses in the `liquidwarelabs.com` domain.

Inventory

This module creates a report listing computer name, computer domain membership, username, processor, memory, display resolution, MAC address, IP address, operating system, Internet Explorer version, and installed applications. The report also lists the service pack version and installed hotfixes.



NOTE: On systems with more than 4GB of RAM, memory is reported incorrectly. On systems with more than one IP address, only the first IP address is listed. Applications must be listed in "Add or Remove Programs" to be included in the report.

Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Action:
Choose whether you want to **Upload XML File** or **Use Reporting Server**.

Upload Path:
The Upload Path designates the UNC name of the network share ProfileUnity will use to store the generated report in XML format. When you create the share, you will need to grant write permission to all users using ProfileUnity. If this share is unavailable or does not have write permission, ProfileUnity will not create the report. This option only applies if you choose **Upload XML File** for the action.

NOTE: Inside the share, ProfileUnity will create an XML file named `"[computer name].xml"` for each client machine that executes this module.

Server URL:
The **Server URL** is the address of the machine running the ProfileUnity Management Console. The URL can either be a fully qualified domain name or an IP address. This option only applies if you choose **Use Reporting Server** for the action.

Example 1:

Filter: No Filter – Apply this to all
Action: Use Reporting Server
Server URL: https://ssv2.liquidwarelabs.com

Example 1 sends inventory reports to the ProfileUnity Management Console located at `https://ssv2.liquidwarelabs.com`. The connection will be made using SSL.

Example 2:

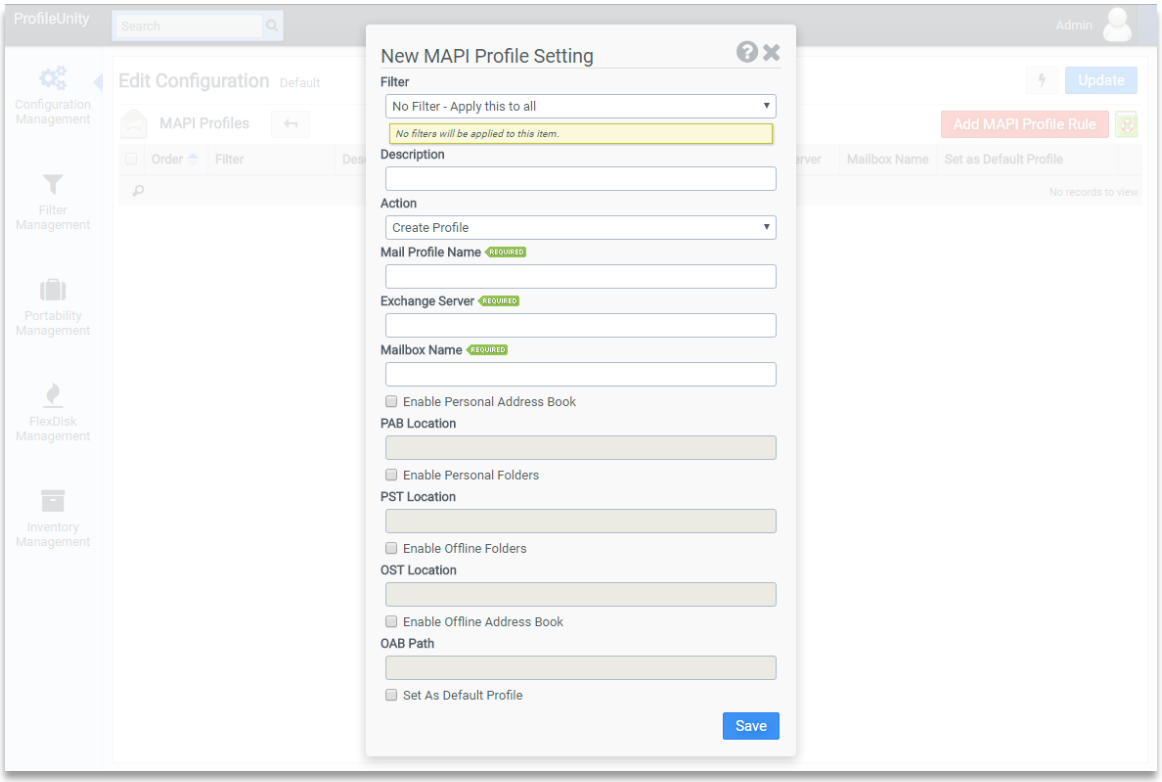
Filter: No Filter – Apply this to all
Action: Upload XML File
Upload Path: \\ATLFS02\inventory$$

Example 2 creates reports in XML format on the network share \\ATLFS02\inventory$.

NOTE: The share name used for the Upload Path is `\\ATLFS02\inventory$`. Since $ is a reserved character, it has been escaped with $$.

MAPI Profiles

This module is used to configure Windows Messaging subsystem profiles for access to Microsoft Exchange Server 2010. The Windows Messaging subsystem is used by Microsoft Outlook to communicate with Microsoft Exchange. This module supports Outlook 2003/2007/2010 in conjunction with Microsoft Exchange 2010. To use this configuration module, go to **Administration > Settings > Miscellaneous** and uncheck the "Hide MAPI Profiles Module".



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Action:
Select one of the following actions:

- **Create Profile**
- **Create - Only if No Profiles Exist**
- **Create - Only if No Profile Exists with this Name**
- **Delete Profile**
- **Delete All Profiles**
- **Delete All Profiles - Except Default Profile**

Mail Profile Name:
Enter the name of the mail profile. This option is not available if **Delete All Profiles** or **Delete All Profiles - Except Default Profile** is selected as the **Action**.

Exchange Server:
Enter the name of the Exchange Server. This option is only available if **Create Profile**, **Create - Only if No Profiles Exist**, or **Create - Only if No Profile Exists with this Name** is selected as the **Action**.

NOTE: Outlook 2003 will hang during logon if the exchange server is not valid or available.

Mailbox Name:
Enter the name of the mailbox. This option is only available if **Create Profile**, **Create - Only if No Profiles Exist**, or **Create - Only if No Profile Exists** with this Name is selected as the **Action**.

Enable Personal Address Book:
Selecting this option will configure the mail profile to use a personal address book. You will need to enter the complete path to the personal address book in the **PAB Location** field. This option is only available if **Create Profile**, **Create - Only if No Profiles Exist**, or **Create - Only if No Profile Exists with this Name** is selected as the **Action**.

PAB Location:
Enter the personal address book location. This option is only available if **Enable Personal Address Book** is selected.

Enable Personal Folders:
Selecting this option will configure the mail profile to use personal folders. You will need to enter the complete path to the personal folder file in the **PST Location** field. This option is only available if **Create Profile**, **Create - Only if No Profiles Exist**, or **Create - Only if No Profile Exists with this Name** is selected as the **Action**.

PST Location:
Enter the personal folder file location. This option is only available if **Enable Personal Folders** is selected.

Enable Offline Folders:
Selecting this option will configure the mail profile to use offline folders. You will need to enter the complete path to the offline folder file in the **OST Location field**. This option is only available if **Create Profile**, **Create - Only if No Profiles Exist**, or **Create - Only if No Profile Exists with this Name** is selected as the **Action**.

NOTE: If the offline folder file specified in OST Location currently exists, it will be deleted. Offline folders are not supported on Terminal Servers.

OST Location:
Enter the offline folder file location. This option is only available if **Enable Offline Folders** is selected.

Enable Offline Address Book:
Selecting this option will configure the mail profile to use an offline address book. You will need to enter the complete path to the offline address book in the **OAB Location field**. This option is only available if **Create Profile**, **Create - Only if No Profiles Exist**, or **Create - Only if No Profile Exists with this Name** is selected as the **Action**.

OAB Location:
Enter the offline address book location. This option is only available if **Enable Offline Address Book** is selected.

Set as Default Profile:
Selecting this option will configure the mail profile as the default profile.

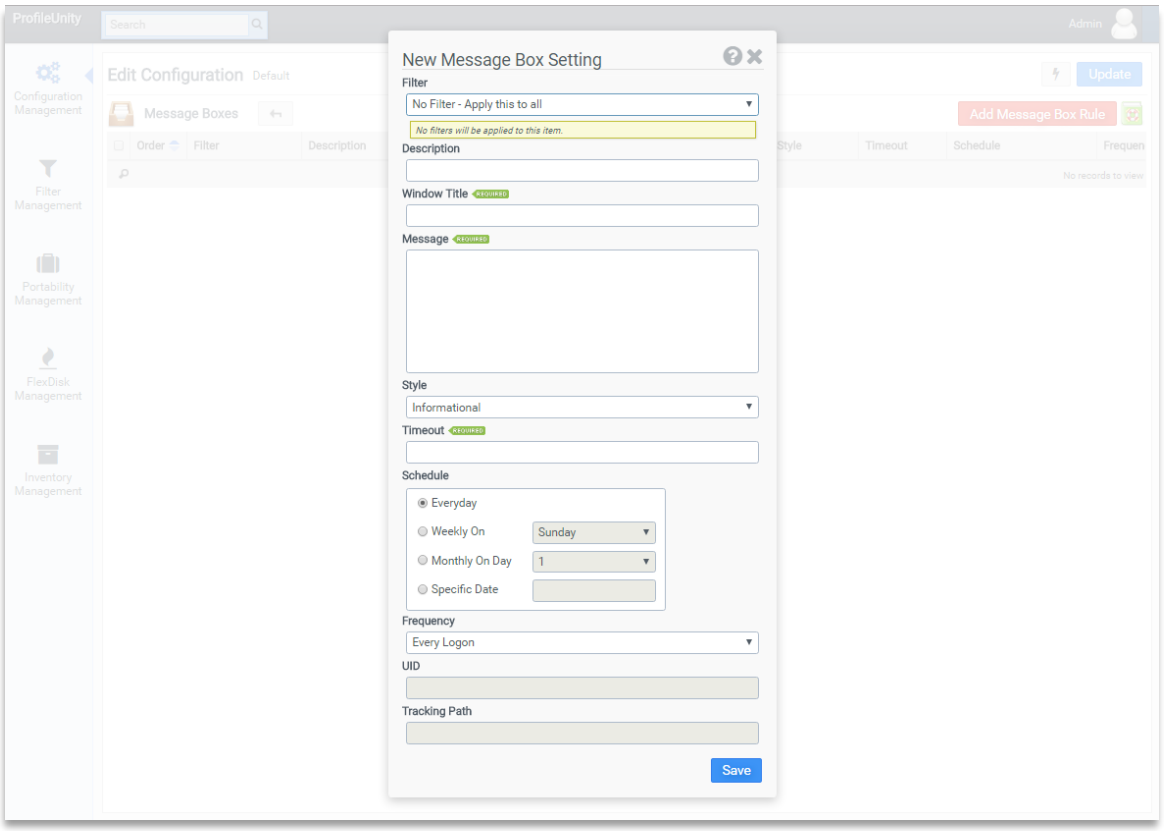
Example:

Filter: No Filter – Apply this to all
Action: Create - Only if No Profiles Exist
Mail Profile Name: LIQUIDWARELABS
Exchange Server: ATLEX01
Mailbox Name: @userid
Enable Personal Address Book: Selected
PAB Location: H:\mail.pab
Enable Personal Folders: Not Selected
Enable Offline Folders: Not Selected
Enable Offline Address Book: Not Selected
Set as Default Profile: Selected

The above example will create a mail profile named `LIQUIDWARELABS` only if no profiles currently exist. This example leverages the macro `@USERID`. When ProfileUnity executes, the `@USERID` macro is expanded into the username. Therefore, the above example will create a mail profile that connects to each user's mailbox on the exchange server named `ATLEX01`. The mail profile will be configured to use `H:\mail.pab` as a personal address book and will be set as the default profile.

Message Boxes

This module allows you to display popup messages to your users during logon.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Window Title:
Enter the text you want displayed as the message box title.

Message:
Enter the text you want displayed in the message box.

Style:
This parameter defines the style of the message box. Choose from:

- **Informational**
- **Warning**
- **Error**

Timeout:
You can configure message boxes to wait for user acknowledgement or to close automatically after a specified amount of time elapses. This parameter represents the number of seconds after which to close the message box. If you do not want the message box to automatically close, enter `0` seconds.

Schedule:
Choose the schedule your message box should display on:

- **Everyday**
- **Weekly and** select a particular day of the week
- **Monthly** and select a particular day of the month
- **Specific Date** and select the date

Frequency:
Choose the frequency your message box should display on:

- **Every Logon**
- **Once Per Day (Computer)**
- **Once Per Day (User)**
- **One Time (Computer)**
- **One Time (User)**

UID:
The **UID** is a unique identifier that is automatically generated by ProfileUnity. Normally, the **UID** does not need to be changed. Changing the **UID** will reset the display history associated with the message box. For example, if you chose a frequency of One Time (User) and you want the message box to display a second time, changing the **UID** will cause the message box to display again. This option does not apply if you choose a **Frequency** of **Every Logon**.

Tracking Path:
The **Tracking Path** designates the UNC name of the network share ProfileUnity will use to store display history. When you create the share, you will need to grant write permission to all users using ProfileUnity. If this share is unavailable or does not have write permission, ProfileUnity will not display the message box. This option does not apply if you choose a **Frequency** of **Every Logon**.

NOTE: Inside the share ProfileUnity will create a folder named after the UID of the message box. Therefore, it is safe to use the same share as the Tracking Path for multiple message boxes. If you wish to reset the execution history for a message box, you can delete the folder associated with the message box's UID.

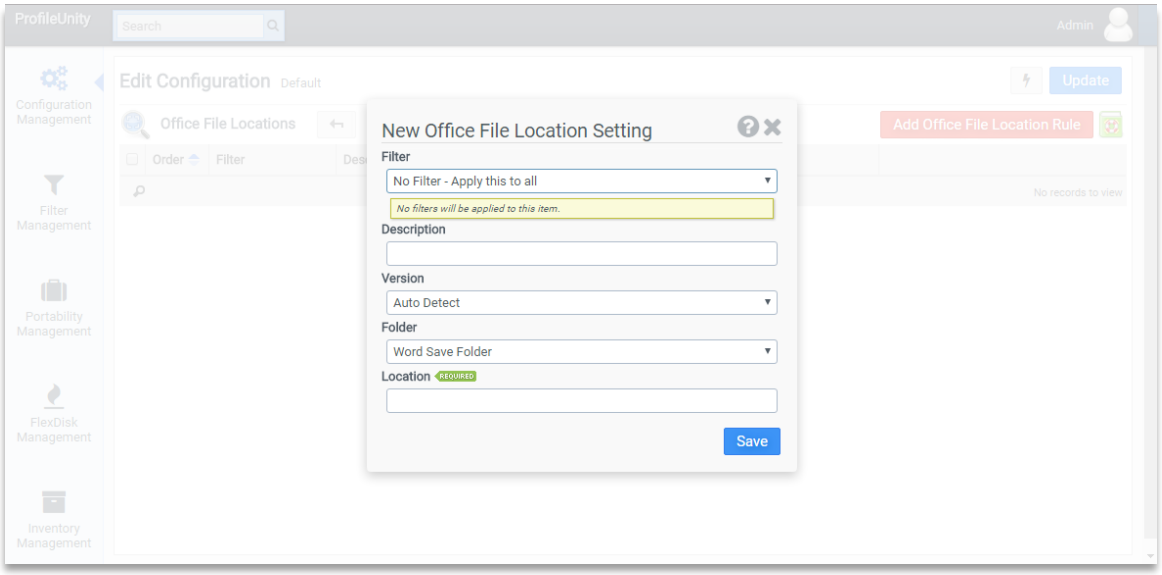
Example:

Filter: No Filter – Apply this to all
Window Title: Weekly Staff Meeting Reminder
Message: The weekly staff meeting is scheduled to take place at 1:30PM in the conference room.
Style: Informational
Timeout: 10
Schedule: Weekly on Monday
Frequency: Every Logon

The above example will display the message box titled `Weekly Staff Meeting Reminder` every Monday at every logon. If the user fails to acknowledge the message box, it will automatically close after being displayed for `10` seconds.

Office File Locations

This module allows you to modify the default file locations for Microsoft Office. This module supports Office 2003/2007/2010/2013/2016.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Version:
Select the version of Office you want to modify. The recommended setting is **Auto Detect**. However, if you want to specify different locations for different versions of Office, you will need to specify the version you want to modify.

Folder:
Select one of the following folders:

- **Word Save Folder** – Supports Word 2003 and Newer.
- **Word Clip Art Folder** – Supports Word 2003 and Newer.
- **Word Tools Folder** – Supports Word 2003 and Newer.
- **Word AutoRecovery Folder** – Supports Word 2003 and Newer.
- **Word Startup Folder** – Supports Word 2003 and Newer.
- **Excel Save Folder** – Supports Excel 2003 and Newer.
- **Excel Alternate Startup Folder** – Supports Excel 2003 and Newer.
- **Power Point Save Folder** – Supports Power Point 2003 and Newer.
- **Access Save Folder** – Supports Access 2003 and Newer.
- **Visio Save Folder** – Supports Visio 2003 and Newer.
- **Publisher Publications Folder** – Supports Publisher 2003 and Newer.
- **Publisher Pictures Folder** – Supports Publisher 2003 and Newer.
- **Office Templates Folder** – Supports Office 2003 and Newer.
- **Office Shared Templates Folder** – Supports Office 2003 and Newer.

Location:
Enter the full path to the location the **Folder** should default to.

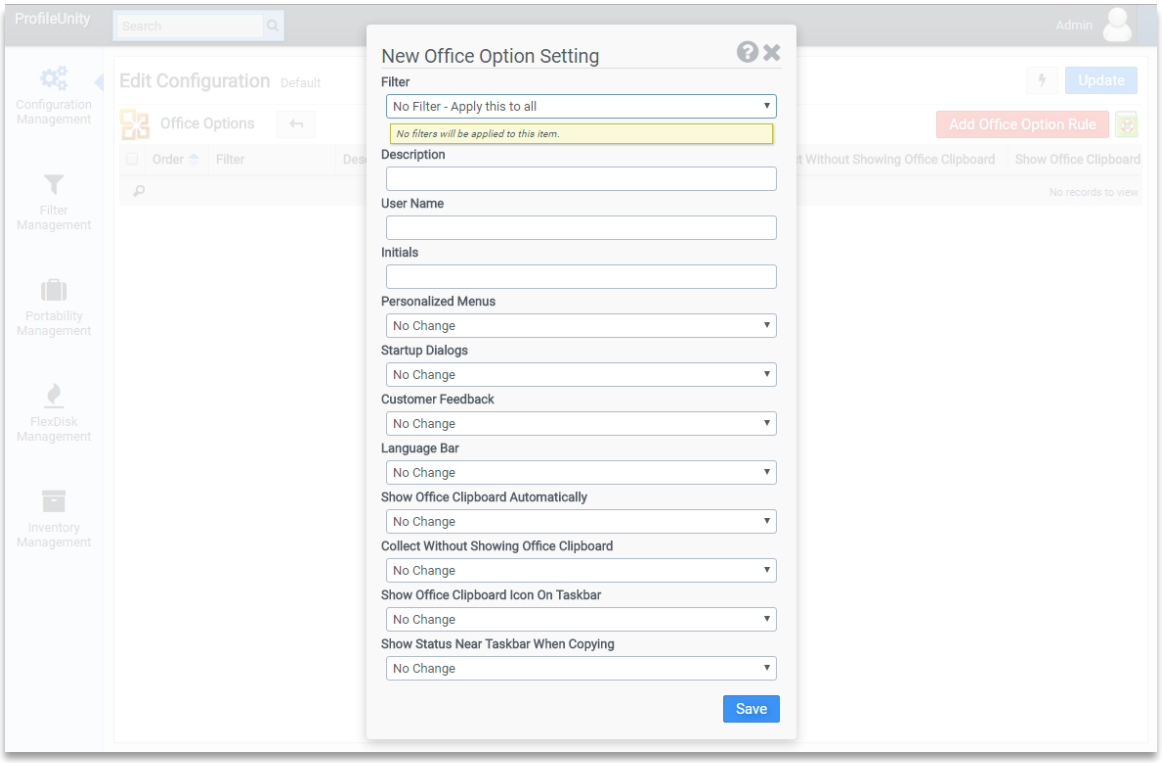
Example:

Filter: No Filter – Apply this to all
Version: Auto Detect
Folder: Word Save Folder
Location: H:\Documents

The above example will redirect the `Word Save Folder` to `H:\Documents`. Since `Auto Detect` has been selected, all installed versions of Word will be modified.

Office Options

This module allows you to configure Microsoft Office user information and set options that control interaction with the user. This module supports Office 2003/2007/2010/2013/2016.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

User Name:
Enter the user name Office should use.

Initials:
Enter the initials Office should use.

Personalized Menus:
Office automatically updates menus to keep the features you use most often at the top of the menu where they are usually more accessible. This option allows you to control the Personalized Menus setting used by your users. Available choices are: **Enable**, **Disable**, and **No Change**.

Startup Dialogs:
This option allows you to control the Getting Started task pane that displays when an Office application is started. Available choices are: **Enable**, **Disable**, and **No Change**. This option supports the following applications from the Office 2003 and newer suites: Access, Excel, Power Point, Word, FrontPage, and Visio. Additionally, this option supports Publisher 2003.

Customer Feedback:
This option controls whether Microsoft is allowed to collect anonymous usage statistics about the way you use Office. Available choices are: **Enable**, **Disable**, and **No Change**.

Language Bar:
This option controls display of the Language Bar on the user desktop. Available choices are: **Enable**, **Disable**, and **No Change**.

Show Office Clipboard Automatically:
This option controls automatic display of the Office clipboard when copying items. Available choices are: **Enable**, **Disable**, and **No Change**.

Collect Without Showing Office Clipboard:
This option allows items to be copied to the Office clipboard without displaying the Office clipboard. Available choices are: **Enable**, **Disable**, and **No Change**.

Show Office Clipboard Icon on Taskbar:
This option controls display of the Office clipboard icon in the system tray when the Office clipboard is active. Available choices are: **Enable**, **Disable**, and **No Change**.

Show Status Near Taskbar When Copying:
This option controls display of the collected item message when copying items to the Office clipboard. Available choices are: **Enable**, **Disable**, and **No Change**.

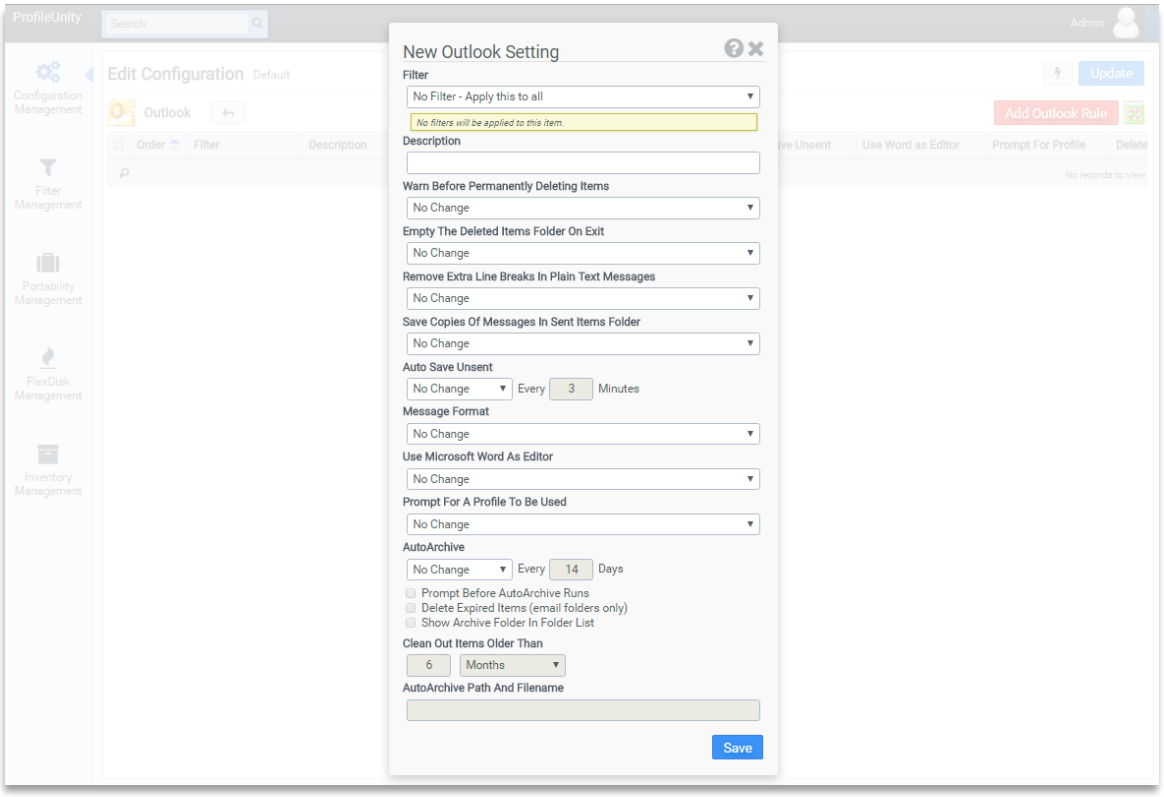
Example:

Filter: No Filter – Apply this to all
User Name: @fullname
Initials: $$first_initial$$last_initial
Personalized Menus: Disable
Startup Dialogs: Disable
Customer Feedback: Disable
Language Bar: Disable
Show Office Clipboard Automatically: Disable
Collect Without Showing Office Clipboard: Enable
Show Office Clipboard Icon on Taskbar: Disable
Show Status Near Taskbar When Copying: Enable

The above example leverages the macro `@FULLNAME` and the User Defined Aliases module. When ProfileUnity executes, the `@FULLNAME` macro is expanded into the user's full name. The `$$first_initial` and `$$last_initial` are User Defined Aliases that expand into the user's first initial and last initial respectively. Therefore, the Office `User Name` field will be set to the user's full name and the Office `Initials` field will be set to the user's first initial and last initial. **Collect Without Showing Office Clipboard** and **Show Status Near Taskbar When Copying** will be enabled. All other available options will be disabled.

Outlook

This module allows you to modify settings that control Microsoft Outlook's behavior for message handling, message format, and AutoArchive.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Warn Before Permanently Deleting Items:
This option controls the display of a warning message prior to deleting items. Available choices are: **Enable**, **Disable**, and **No Change**.

Empty the Deleted Items folder on exit:
When this option is enabled, Outlook will delete items in the Deleted Items folder when closed. Available choices are: **Always**, **Never**, **On Sunday**, **On Monday**, **On Tuesday**, **On Wednesday**, **On Thursday**, **On Friday**, **On Saturday**, and **No Change**.

Remove extra line breaks in plain text messages:
When this option is enabled, Outlook will remove extra line breaks in plain text messages. Available choices are: **Enable**, **Disable**, and **No Change**. This option supports Outlook 2003 and newer versions.

Save copies of messages in Sent Items folder:
When this option is enabled, Outlook will save copies of sent messages in the Sent Items folder. Available choices are: **Enable**, **Disable**, and **No Change**.

AutoSave Unsent:
This option controls when Outlook saves unsent items. Available choices are: **Enable**, **Disable**, and **No Change**. If you enable this option, you will need to specify the number of minutes.

Message Format:
This option controls message format used by Outlook. Available choices are: **HTML**, **Rich Text**, **Plain Text**, and **No Change**.

Use Microsoft Word as editor:
When this option is enabled, Outlook will use Microsoft Word as the message editor. Available choices are: **Enable**, **Disable**, and **No Change**. This option supports Outlook 2003.

Prompt for a profile to be used:
When this option is enabled, Outlook will prompt the user to choose a profile on startup. Available choices are: **Enable**, **Disable**, and **No Change**.

AutoArchive:
This option controls when Outlook AutoArchives items. **Available choices are: Enable, Disable, and No Change**. If you enable this option, you will need to specify the number of days.

Prompt before AutoArchive runs:
When this option is selected, the user will be notified prior to Outlook AutoArchiving items.

Delete expired items (e-mail folders only):
When this option is selected, Outlook will delete expired items instead of archiving them when AutoArchive runs.

NOTE: If you select this option, items will be permanently deleted, not archived.

Show archive folder in folder list:
When this option is selected, the AutoArchive file will be displayed in the Outlook folder list. This option supports Outlook 2003 and newer versions.

Clean out items older than:
This option specifies the default time period Outlook uses when AutoArchiving. Items older than this time period will be AutoArchived. This option supports Outlook 2003 and newer versions.

AutoArchive Path and Filename:
Enter the full path to the AutoArchive file. If you leave this option blank, Outlook will use the user's currently configured AutoArchive file.

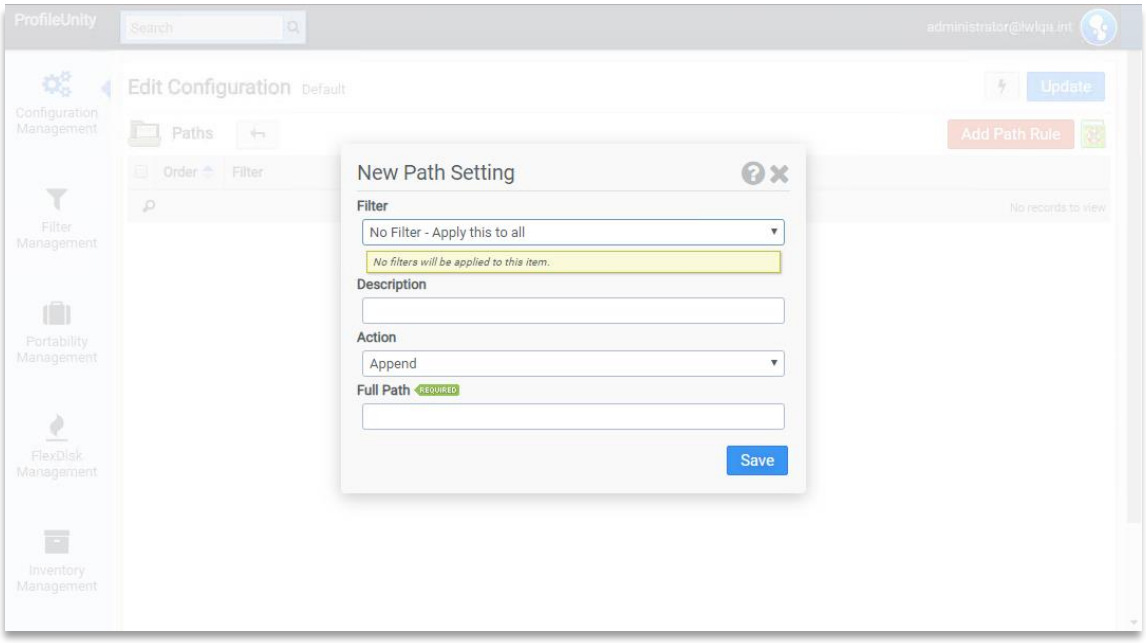
Example:

Filter: No Filter – Apply this to all
Warn Before Permanently Deleting Items: Enable
Empty the Deleted Items folder on exit: On Fri
Remove extra line breaks in plain text messages: Disable
Save copies of messages in Sent Items folder: Enable
AutoSave unsent every ___ minutes: Enable 5
Message Format: Rich Text
Use Microsoft Word as editor: Disable
Prompt for a profile to be used: Disable
AutoArchive Every ___ Days: No Change

The above example enables **Warn Before Permanently Deleting Items** and **Save copies of messages in Sent Items folder**. Every Friday, Outlook will be configured to **Empty the Deleted Items folder on exit**. Additionally, Outlook will AutoSave unsent messages every `5` minutes and messages will be composed in `Rich Text` format. The user's current AutoArchive settings will be preserved. All other available options will be disabled.

Paths

This module allows you to make modifications to the search path. The path is modified in the environment of the current user.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Action:
Select one of the following actions:

- **Append** – Adds to the current path
- **Overwrite** – Replaces the current path

NOTE: The append action checks if the new path is already part of the configured path prior to adding it.

Full Path:
Enter the path you want to append or replace the configured path with. Surrounding quotes are automatically added to the path.

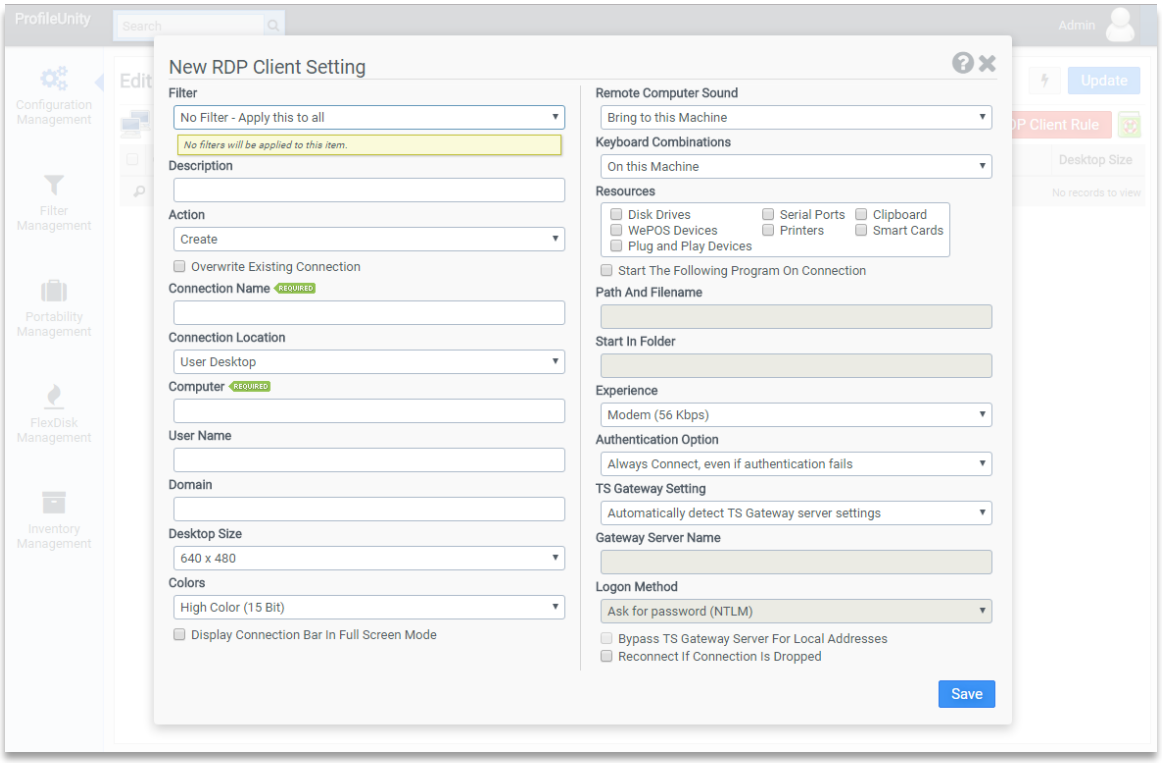
Example:

Filter: No Filter – Apply this to all
Action: Append
Path: C:\Program Files\Tools

The above example will append `C:\Program Files\Tools` to the configured path.

RDP Client

This module allows you to configure RDP clients from a centralized location. This module supports RDP versions 5.1 (for Windows XP Client) and higher.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Action:
Select one of the following actions:

- **Create** – Creates a new RDP connection.
- **Delete** – Deletes an existing RDP connection.

NOTE: If a client machine has more than one version of the RDP client installed, precedence is given to the newest version. For example, if a client machine has both RDP 4.0 and RDP 5.2 installed, connections will be created and deleted for the RDP 5.2 version.

Overwrite Existing Connection:
When this option is selected, existing RDP connections will be overwritten. This option is only available if **Create** is selected as the **Action**.

Connection Name:
Enter the name of the RDP connection you want to create or delete. Subfolders can be specified by entering the relative path followed by the RDP connection name. Surrounding quotes are automatically added to the name.

Connection Location:
Select the location where the RDP connection should be created or deleted. Choose from:

- **User Desktop**
- **User Favorites**
- **User Start Menu**
- **User Programs Group**
- **User Startup Group**
- **User Send To**
- **User Quick Launch Bar**

Computer:
Enter the name of the computer you want the RDP client to connect to. This option is only available if **Create** is selected as the **Action**.

User Name:
Enter the user name the RDP client should use when connecting to the server. This option is only available if **Create** is selected as the **Action**.

Domain:
Enter the domain name the RDP client should use when connecting to the server. This option is only available if **Create** is selected as the **Action**.

Desktop Size:
Select the desktop size the RDP client should use. When the **80% of Screen** choice is selected, the RDP desktop size is derived from the client resolution. This option is only available if **Create** is selected as the **Action**.

Client Resolution	RDP Desktop Size
640 x 480	640 x 480
800 x 600	640 x 480
1024 x 768	800 x 600
1280 x 1024	1024 x 768
1600 x 1200	1280 x 1024
80% of Screen	Based on Client
Full Screen	Based on Client

Colors:
Select the color depth you want the RDP client to use. This option applies to RDP version 5.1 and newer and is only available if **Create** is selected as the **Action**. Choose from:

- **High Color (15 bit)**
- **High Color (16 bit)**
- **True Color (24 bit)**
- **Highest Color (32 bit)**

Display connection bar in full screen mode:
When this option is selected, the connection bar will be displayed across the top of the display when the RDP connection is in full screen mode. This option applies to RDP version 5.1 and newer and is only available if **Create** is selected as the **Action**.

Remote Computer Sound:
Select how you want the RDP client to handle sound. This option applies to RDP version 5.1 and newer and is only available if **Create** is selected as the **Action**. Choose from:

- **Bring to this Machine**
- **Do Not Play**
- **Leave at Remote Machine**

Keyboard Combinations:
Select how you want the RDP client to handle keyboard combinations such as ALT+TAB. This option applies to RDP version 5.1 and newer and is only available if **Create** is selected as the **Action**. Choose from:

- **On this Machine**
- **On Remote Machine**
- **Only in Full Screen Mode**

Resources:
This option controls the resources that are available to the RDP client. This option applies to RDP version 5.1 and newer and is only available if **Create** is selected as the **Action**. Additionally, **Clipboard, Plug and Play Devices,** and **WePOS Devices** only apply to RDP version 6.0 and newer.

Start the Following Program on Connection:
Selecting this option instructs the RDP client to start a program on connection. This option is only available if **Create** is selected as the **Action**.

Path and Filename:
Enter the full path and filename of the program the RDP client should execute when connecting to the server. This option is only available if **Start the Following Program on Connection** is selected and **Create** is selected as the **Action**.

Start in Folder:
Enter the full path of the folder the RDP client should start in when executing a program on connection to the server. This option is only available if **Start the Following Program on Connection** is selected and **Create** is selected as the **Action**.

Experience:
Select the connection speed you want the RDP client to be optimized for. For RDP version 4.0 and 5.0, selecting **Modem (28.8 Kbps)** or **Modem (56 Kbps)** enables the Low Speed Connection option in the RDP client. This option is only available if **Create** is selected as the **Action**. Choose from:

- **Modem (56 Kbps)**
- **Low-speed broadband (256 Kbps – 2 Mbps)**
- **Satellite (2 Mbps – 16 Mbps with high latency)**
- **High-speed broadband (2 Mbps – 10 Mbps)**
- **WAN (10 Mbps or higher with high latency)**
- **LAN (10Mbps or higher)**
- **Detect connection quality automatically**

Authentication Option:
Select the server authentication level you want the RDP client to use. This option applies to RDP version 6.0 and newer and is only available if **Create** is selected as the **Action**. Choose from:

- **Always Connect, even if authentication fails**
- **Do not connect if authentication fails**
- **Warn me if authentication fails**

TS Gateway Setting:
Select the gateway setting you want the RDP client to use. This option applies to RDP version 6.0 and newer and is only available if **Create** is selected as the **Action**. Choose from:

- **Automatically detect TS Gateway server settings**
- **Use these TS Gateway server settings**
- **Do not use a TS Gateway server**

Gateway Server Name:
Enter the server name of the TS Gateway Server you want the RDP client to use. This option applies to RDP version 6.0 and newer and is only available if **Use these TS Gateway server settings** is selected and **Create** is selected as the **Action**.

Logon Method:
Select the logon method you want the RDP client to use. This option applies to RDP version 6.0 and newer and is only available if **Use these TS Gateway server settings** is selected and **Create** is selected as the **Action**. Choose from:

- **Ask for password (NTLM)**
- **SmartCard**
- **Allow me to select later**

Bypass TS Gateway server for local addresses:
Selecting this option instructs the RDP client to bypass the TS Gateway server when the destination is local. This option applies to RDP version 6.0 and newer and is only available if **Use these TS Gateway server settings** is selected and **Create** is selected as the **Action**.

Reconnect if Connection is Dropped:
Selecting this option instructs the RDP client to attempt to automatically reconnect if the connection is dropped. This option applies to RDP version 5.2 and is only available if **Create** is selected as the **Action**.

Example 1:

Filter: No Filter – Apply this to all
Action: Delete
Connection Name: Accounting\Solomon
Connection Location: User Start Menu

Example 1 deletes the RDP connection named `Solomon` located in the `Accounting` folder in the `User Start Menu`.

Example 2:

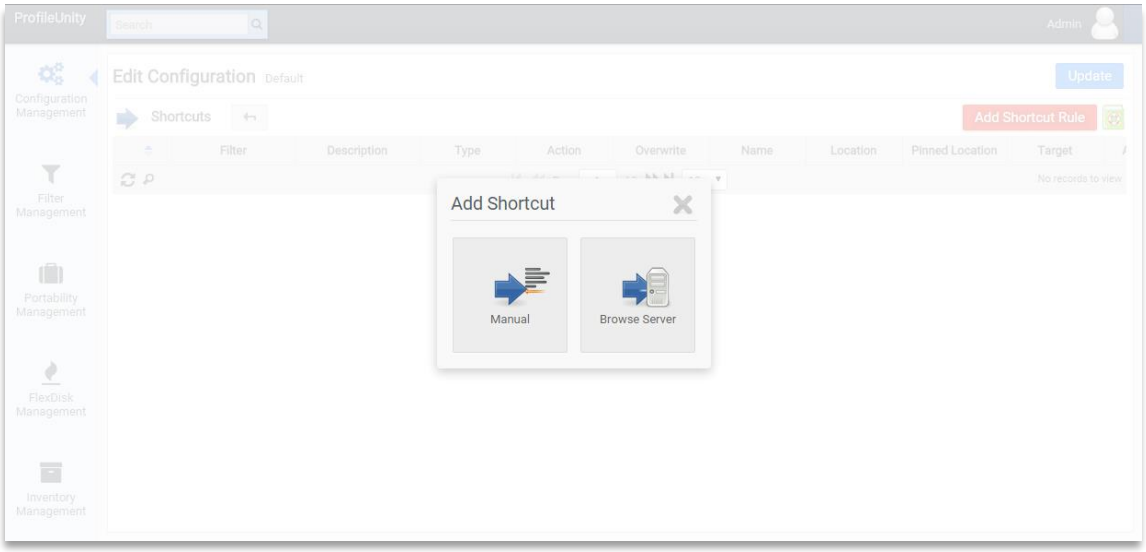
Filter: No Filter – Apply this to all
Action: Create
Overwrite Existing Connection: Not Selected
Connection Name: Human Resources\Abra Suite
Connection Location: User Start Menu
Computer: ATLHR
User Name: @userid
Domain: MyDomain
Desktop Size: 80% of Screen
Colors: High Color (16 Bit)
Display connection bar in full screen mode: Not Selected
Remote Computer Sound: Bring to this Computer
Keyboard Combinations: On the Local Computer
Automatically Connect Disk Drives: Not Selected
Automatically Connect Printers: Not Selected
Automatically Connect Serial Ports: Not Selected
Automatically Connect Smart Cards: Not Selected
Start the Following Program on Connection: Selected
Path and Filename: C:\Program Files\Best Software\Abra Suite\Programs\Abra32.exe
Start in Folder: C:\Program Files\Best Software\Abra Suite\Programs
Experience: Broadband (128 Kbps - 1.5 Mbps)
Authentication Option: Always connect, even if authentication fails
TS Gateway Setting: Automatically detect TS Gateway server settings
Reconnect if Connection is Dropped: Selected

Example 2 creates a RDP connection named `Abra Suite`. The connection will be created in the `User Start Menu` in a subfolder named `Human Resources`. If the connection already exists, it will not be overwritten. This example leverages the macro `@USERID`. When ProfileUnity executes, the `@USERID` macro is expanded into the user's logon name. Therefore, the created RDP connection will connect to server `ATLHR`, use the user's logon name as the user name, use domain name `MyDomain`, use `80% of Screen` for the desktop size, and use `High Color (16 Bit)` color depth. Remote Computer Sound will be brought to the local computer. Keyboard Combinations will stay on the local computer. The RDP connection will start `C:\Program Files\Best Software\Abra Suite\Programs\Abra32.exe` and start in the `C:\Program Files\Best Software\Abra Suite\Programs` folder when double clicked. This connection will be optimized for a `Broadband (128 Kbps - 1.5 Mbps)` network and the connection will automatically reconnect if dropped.

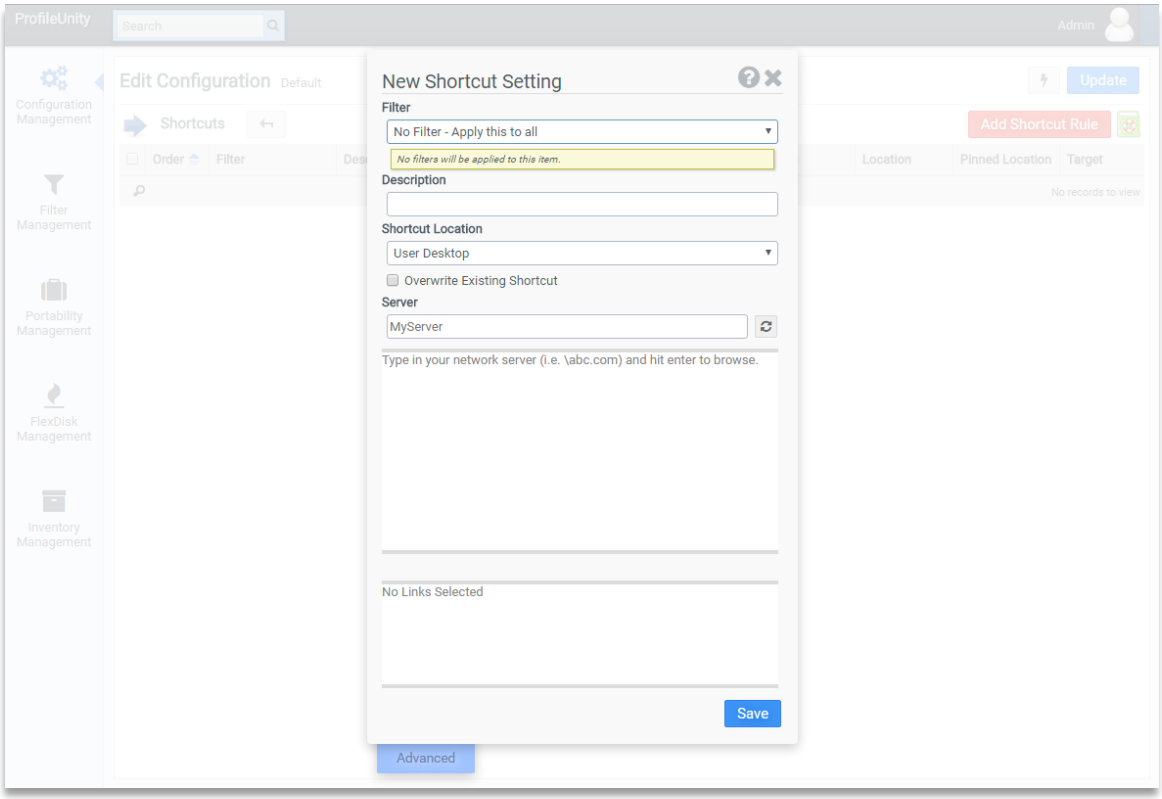
Shortcuts

This module allows you to create shell shortcuts and web links.

Choose whether to add a shortcut manually or browse the server to select multiple shortcuts to be added at one time.

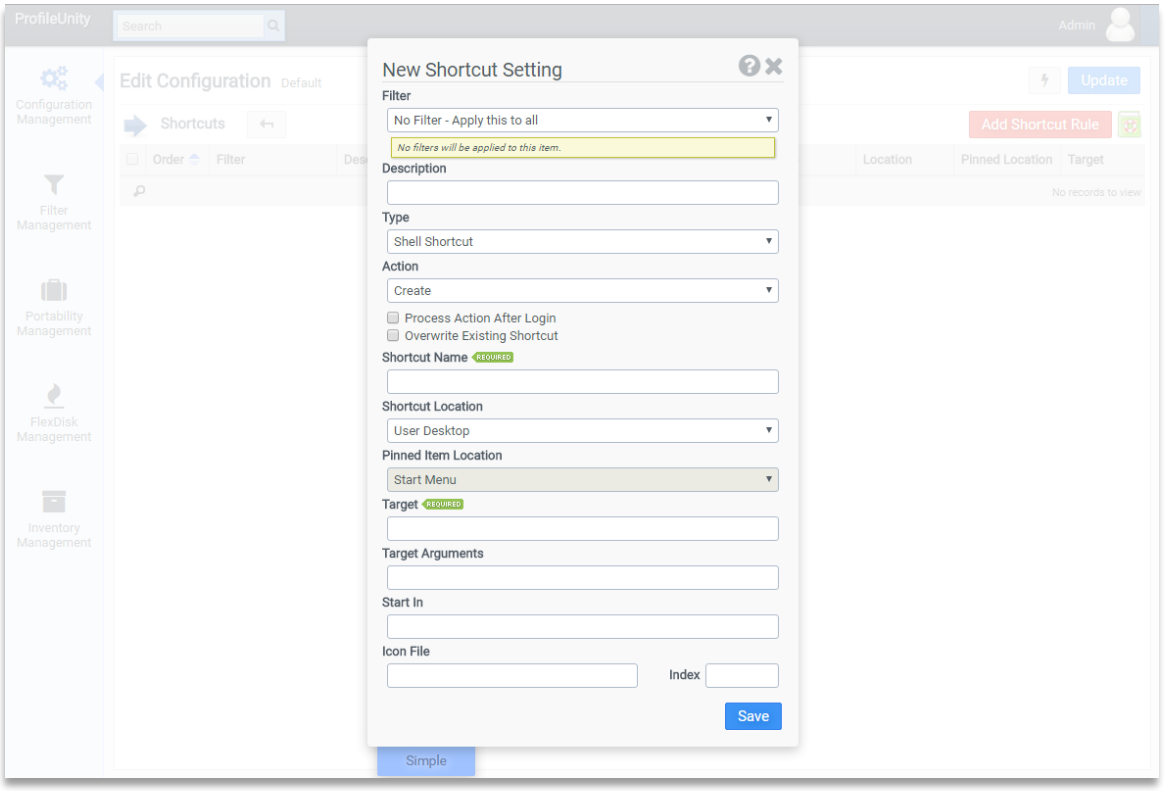


Select **Browse Server** and then type your server name and press **Enter** to search a server for shortcuts.



Select **Manual** to setup shortcut settings manually.

All options for the Advanced settings are shown below.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Type:
Select the type of shortcut you want to create or delete. Shell shortcuts are pointers to programs and files on your computer. Web links are used by Internet Explorer for accessing sites on the Internet. Choose from:

- **Shell Shortcut**
- **Web Link**
- **Pinned Item**

Action:
Select one of the following actions:

- **Create** – Creates a new shortcut.
- **Delete** – Deletes an existing shortcut.
- **Delete All** – Deletes all existing shortcuts.

Process Action After Login
Indicates whether the action should occur before or after the user has logged in.

Overwrite Existing Shortcut:
When this option is selected, existing shortcuts will be overwritten. This option is only available if **Create** is selected as the **Action**.

Shortcut Name:
Enter the name of the shortcut you want to create or delete. Subfolders can be specified by entering the relative path followed by the shortcut name. Surrounding quotes are automatically added to the name.

Shortcut Location:
Select the location where the shortcut should be created or deleted:

- **User Desktop**
- **User Favorites**
- **User Start Menu**
- **User Programs Group**
- **User Startup Group**
- **User Send To**
- **User Quick Launch Bar**
- **User Links**
- **All Users Desktops**
- **All Users Start Menu**

Pinned Item Location:
Choose the location where the pinned item should be created or deleted if **Pinned Item** was selected for the **Type**:

- **Start Menu**
- **Taskbar**
- **Quick Access**

Target:
When creating a Shell Shortcut, enter the full path to the shortcut target. When creating a Web Link, enter the URL you want the link to resolve to. Surrounding quotes are automatically added to the target. This option is only available if **Create** is selected as the **Action**.

NOTE: When creating a Shell Shortcut, you must specify the full path in the target. Partial paths are not supported. When creating web links, you must specify the full URL in the target. A full URL starts with http://, https://, or ftp://.

For Modern Apps or apps installed from the Windows Store, you will need to provide the Application User Model ID (AUMID) rather than the full path of the desktop application executable. For example, the AUMID for Skype is `Microsoft.SkypeApp_kzf8qxf38zg5c!App`. The AUMID can be found several ways including using the Registry or PowerShell. When using the Registry, search for the `AppUserModelID`. When using Windows PowerShell, run the following script to list the AUMIDs installed on a desktop:

```
$installedapps = get-AppxPackage
foreach ($app in $installedapps)
{
    foreach ($id in (Get-AppxPackageManifest $app).package.applications.application.id)
    {
        $line = $app.Name + " = " + $app.packagefamilyname + "!" + $id
        echo $line

    }
}
```

Target Arguments:
Enter any arguments required by the shortcut target. Target Arguments are not automatically quoted. This option is only available if **Shell Shortcut** or **Pinned Item** is selected as the **Type** and **Create** is selected as the **Action**.

Start In:
Enter the working directory the shortcut target should start in. Surrounding quotes are automatically added to the start in directory. This option is only available if **Shell Shortcut** or **Pinned Item** is selected as the **Type** and **Create** is selected as the **Action**.

Icon File:
Enter the full path to the icon the shortcut should use. Surrounding quotes are automatically added to the icon file. This option is only available if **Create** is selected as the **Action**.

Index:
If the shortcut icon file contains more than one icon, enter the numerical index of the icon the shortcut should use. This option is only available if **Create** is selected as the **Action**.

Example 1:

Filter: No Filter – Apply this to all
Type: Shell Shortcut
Action: Create
Overwrite Existing Shortcut: Selected
Shortcut Name: Finance\Tax Tables
Shortcut Location: User Start Menu
Target: P:\Finance\Tax Tables.xls
Start In: P:\Finance

Example 1 creates a shell shortcut named `Tax Tables`. The shortcut will be created in the `User Start Menu` in a subfolder named Finance. If the shortcut already exists, it will be overwritten. When double clicked, the shortcut will open `P:\Finance\Tax Tables.xls` in Microsoft Excel.

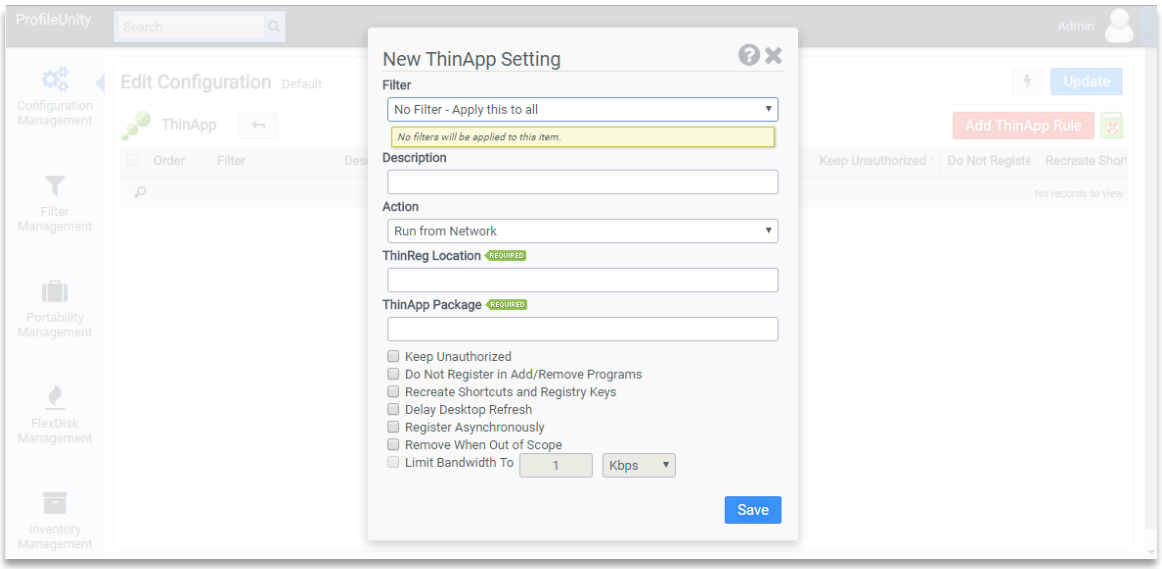
NOTE: Missing subfolders are created automatically. In the above example, the Finance folder will be created if it does not already exist.

Example 2:
Filter: No Filter – Apply this to all
Type: Web Link
Action: Create
Overwrite Existing Shortcut: Not Selected
Shortcut Name: Company Intranet
Shortcut Location: User Desktop
Target: http://int.mycompany.com

Example 2 creates a web link named `Company Intranet` on the `User Desktop`. If the shortcut exists, it is not overwritten. When double clicked, the shortcut will open the website `http://int.mycompany.com` in Internet Explorer.

ThinApp

This module allows you to dynamically provision ThinApp packages.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Action:
Select one of the following actions:

- **Run from Network** - The ThinApp package is executed from the network location.
- **Run from Local Cache** - The ThinApp package is cached on and runs from the local machine.
- **Remove** - The ThinApp package is removed and the local cache is purged.

ThinReg Location:
Enter the full path to the ThinReg executable.

ThinApp Package:
Enter the full path to the ThinApp package(s). Wildcards are supported.

Keep Unauthorized:
Enabling this option passes the /k switch to ThinReg.

Don't Register in Add/Remove Programs:
Enabling this option passes the /noarp switch to ThinReg.

Re-create Shortcuts and Registry Keys:
Enabling this option passes the /r switch to ThinReg.

Delay Desktop Refresh:
Enabling this option passes the /nodesktoprefresh switch to ThinReg. This option requires ThinReg version 6.6.1 or newer.

Register Asynchronously:
By default, ProfileUnity waits for each ThinReg operation to complete prior to continuing. When this option is selected, ProfileUnity will not wait and will process multiple ThinReg operations in parallel.

Remove when Out of Scope:
Enabling this option will remove the ThinApp package if the selected Filter returns false.

Limit Bandwidth to:
Enabling this option will limit the amount of bandwidth used while caching the ThinApp package on the local machine. This option is only available if **Run from Local Cache** is chosen for **Action**.

Example 1:

Filter: No Filter – Apply this to all
Action: Run from Network
ThinReg Location: \\file01\thinapps\thinreg.exe
ThinApp Package: \\file01\thinapps.exe*
Keep Unauthorized: Not Selected
Don't Register in Add/Remove Programs: Not Selected
Re-create Shortcuts and Registry Keys: Not Selected
Remove when Out of Scope: Not Selected

Example 1 will provision all the ThinApp packages located in `\\file01\thinapps` on the local machine. Each ThinApp package will be executed from the network.

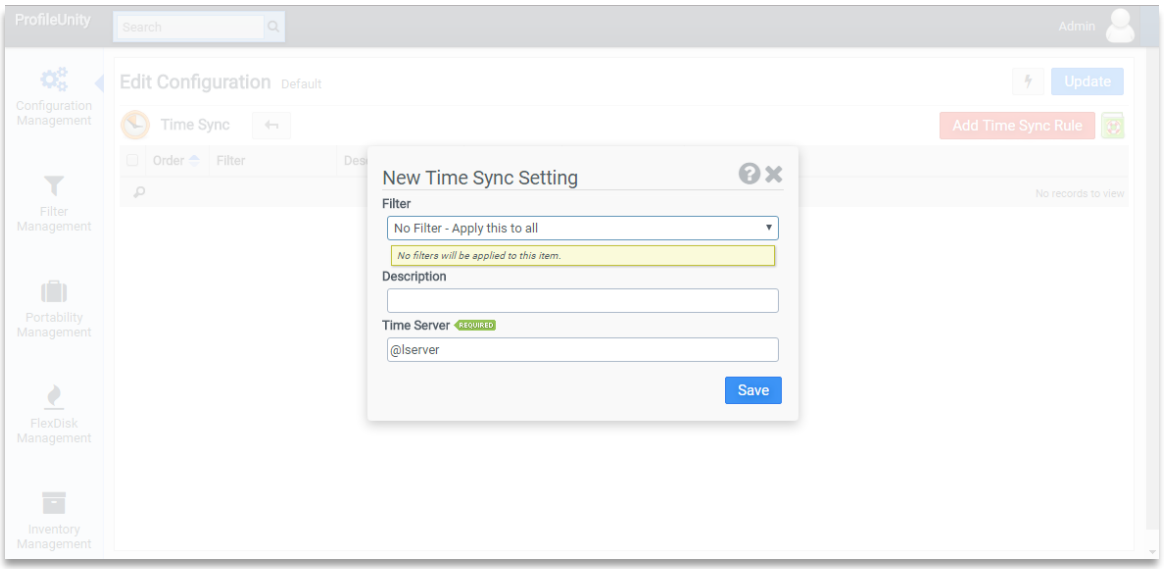
Example 2:

Filter: No Filter – Apply this to all
Action: Run from Local Cache
ThinReg Location: \\file01\clientapps\thinreg.exe
ThinApp Package: \\file01\clientapps\Firefox.exe
Keep Unauthorized: Not Selected
Don't Register in Add/Remove Programs: Not Selected
Re-create Shortcuts and Registry Keys: Not Selected
Remove when Out of Scope: Selected
Limit Bandwidth to: 1 Mbps

Example 2 provisions the ThinApp package `\\file01\clientapps\Firefox.exe` on the local machine. The ThinApp package will be cached locally using 1 `Mbps` of bandwidth and will execute from the local cache. Since `Remove` when Out of Scope has been selected, if the filter Laptop Users returns false, the ThinApp package will be removed.

Time Sync

This module allows you to set the time on your client machines from a centralized source. This feature is designed for use with Microsoft clients lacking a native time service. On clients with a native time service, the preferred method for setting the client machine's time is to use the time service.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Time Server:
Enter the UNC name of the time server the client should set its time from.

NOTE: The recommended setting for the time server is the macro @LSERVER. The @LSERVER macro expands during logon into the name of the domain controller processing the logon request. If you use an alternative time server, you will need to use the server's UNC name. For example, if you wanted to use a server name SVRA1, you would need to enter \\SVRA1 as the time server.

Example:

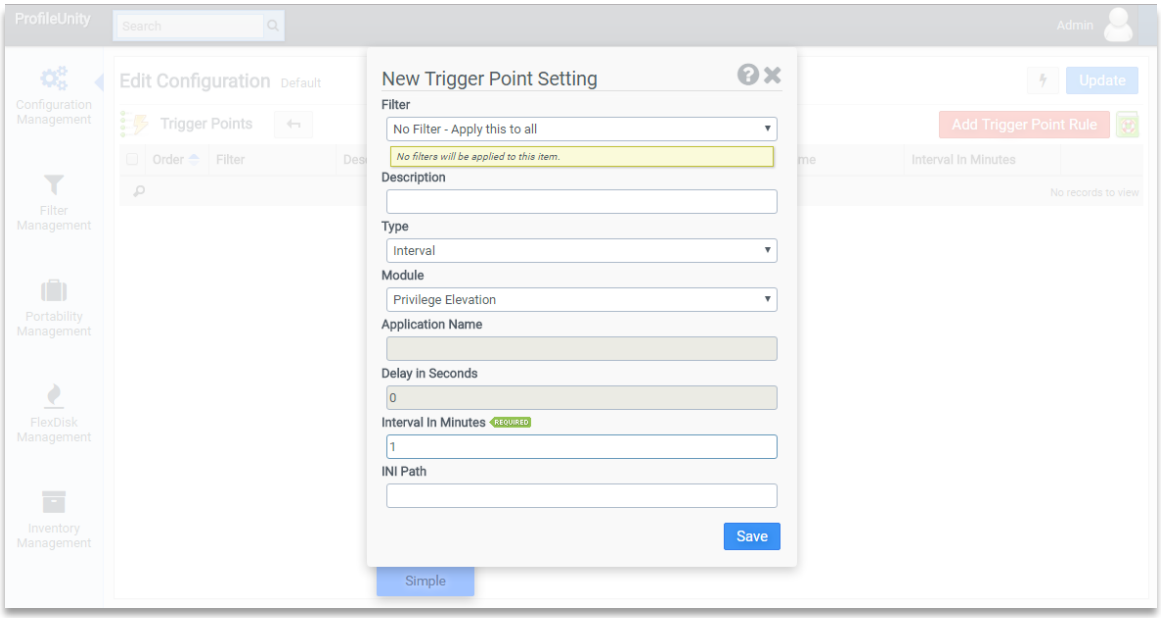
Filter: No Filter – Apply this to all
Time Server: @lserver

The above example synchronizes the client machine's clock with the time obtained from the domain controller processing the logon request.

NOTE: Setting the system time on a Windows machine requires "Change the system time" right. By default, users do not have this right.

Trigger Points

This module allows other modules to run based on various events that occur on the machine. Trigger Points applies to Windows 7 and higher or Windows Server 2008 R2 and higher machines only.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Type:
Select the type of event to listen to. Choose from:

- **Interval**
- **PCoIP Reconnect**
- **PCoIP Connect**
- **PCoIP Disconnect**
- **ICA Connect**
- **ICA Disconnect**
- **RDP Connect**
- **RDP Disconnect**
- **Desktop Unlock**
- **Desktop Lock**
- **IP Address Change**
- **Application Open**
- **Application Close**
- **Computer Shutdown**
- **Post Login**

Module:
Select any of the ProfileUnity configuration modules you would like to run except for the Main module.

Application Name:
The application name to monitor for on **Application Open** or **Application Close** events.

Delay in Seconds:
Set how long to delay before triggering the module with the following events: **IP Address Change** and **Computer Shutdown**. For all other events, the delay is set to 0.

Interval in Seconds:
Enter the interval rate at which to trigger the module with the **Interval** event. For all other events, the interval is set to 15 seconds.

INI Path:
INI file is an advanced setting. By default, the current INI is assumed to process for trigger points.

Example 1:

Filter: No Filter – Apply this to all
Type: RDP Connect
Module: Privilege Elevation
INI Path:

The above example creates a trigger that will listen for any RDP Connects on a machine. Upon a RDP Connect, the Privilege Elevation module will be triggered.

Example 2:

Filter: No Filter – Apply this to all
Type: PCoIP Reconnect
Module: Printers
INI Path:

Example 2 creates a trigger that will run the Printers module on a PCoIP reconnect. Now when a user moves around the building, they get a new default printer based on their current location.

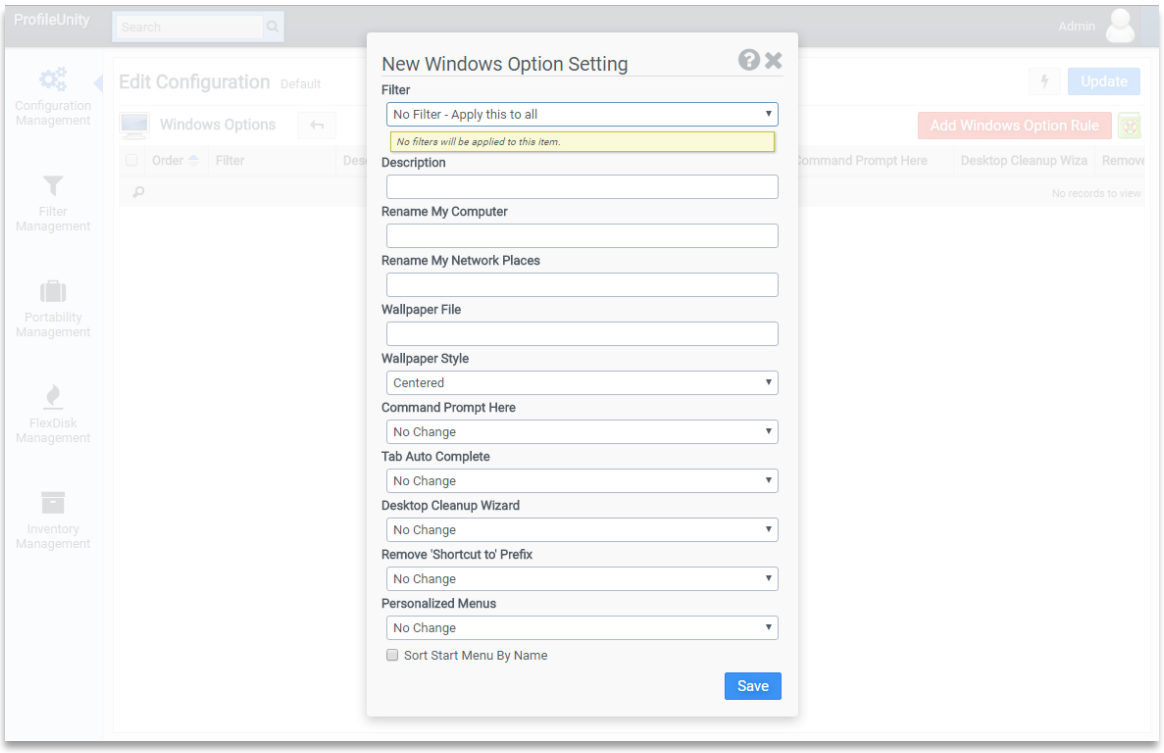
Example 3:

Filter: No Filter – Apply this to all
Type: Interval
Module: Portability Save
Interval in Minutes: 60
INI Path:

Example 3 creates a trigger that will save user portability settings every hour instead of just at logoff.

Windows Options

This module allows you to modify settings that control the Windows user interface.



Filter:
Select the name of the filter you want assigned to this configuration element.

Description:
Enter a description for this rule.

Rename My Computer:
Enter the name you want My Computer renamed to.

NOTE: The recommended setting is the macro @WKSTA. The @WKSTA macro expands into the name of the computer during user logon.

Rename My Network Places:
Enter the name you want My Network Places renamed to.

Wallpaper File:
Enter the full path to the bitmap file Windows should use as desktop wallpaper.

NOTE: The wallpaper file must be a bitmap.

Wallpaper Style:
Choose the style of your wallpaper: **Tiled**, **Centered**, or **Stretched**.

Command Prompt Here:
This option enables you to right click a folder in Windows Explorer and choose Command Prompt Here. This action opens a Command Prompt window with the selected folder as your active directory. Available choices are: **Enable, Disable, and No Change**.

Tab Auto Complete:
This option controls command prompt auto complete functionality. When enabled, you can start typing the name of a file in a command prompt and then press TAB. The shell will attempt to complete the name of the file for you. Available choices are: **Enable, Disable, and No Change**.

Desktop Cleanup Wizard:
The Desktop Cleanup Wizard displays a list of the desktop icons that have not been used for 60 days or more, enabling you to remove those icons that you don't want on your desktop. This option allows you to control the Desktop Cleanup Wizard setting used by your users. Available choices are: **Enable, Disable, and No Change**. This option is only available on Windows XP.

Remove "Shortcut To" Prefix:
This option allows you to remove the "Shortcut to" text prefix on your shortcuts. Available choices are: **Enable, Disable, and No Change**.

Personalized Menus:
On Windows clients, the start menu is automatically updated to keep the items you use most often at the top of the menu where they are usually more accessible. This option allows you to control the Personalized Menus setting used by your users. Available choices are: **Enable, Disable, and No Change**.

Sort Start Menu by Name:
Selecting this option will sort the Start Menu alphabetically.

Example:

Filter: No Filter – Apply this to all
Rename My Computer: @wksta
Rename My Network Places:
Wallpaper File: P:\MyCompany.bmp
Command Prompt Here: Enable
Tab Auto Complete: Enable
Desktop Cleanup Wizard: Disable
Remove "Shortcut To" Prefix: No Change
Personalized Menus: Disable
Sort Start Menu by Name: Selected

The above example renames My Computer to the computer's name, sets the wallpaper file to `P:\MyCompany.bmp`, enables Command Prompt Here, enables Tab Auto Complete, disables the Desktop Cleanup Wizard, disables Personalized Menus, and sorts the Start Menu alphabetically.

NOTE: Leaving a value blank will preserve the current setting. In the above example, My Network Places will not be renamed.

Appendix D - Custom Functions (Deprecated Feature)

The Custom Function filter condition allows filter logic to be extended through the use of a custom written KiXtart function. There are 3 steps required to setup and use this feature.

Step 1: Create Your Custom Function

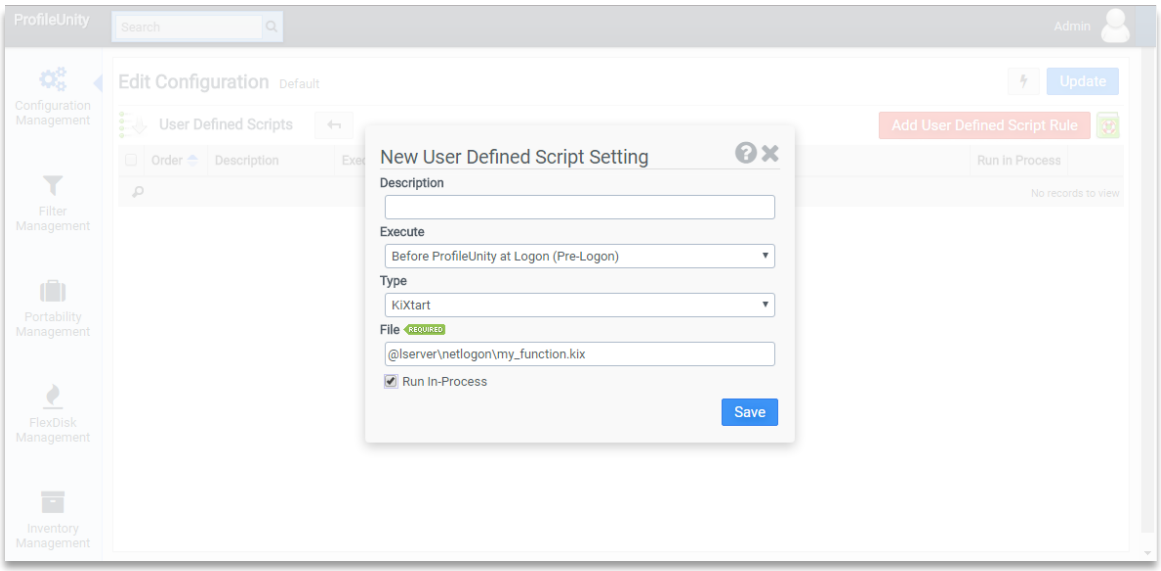
Write a custom KiXtart function that returns 1 on success and 0 on failure. Save this function into a file located in the NETLOGON share on your domain controller.

Example:

```
function demo($var, $val)
    if ($var = $val)
        ; return 1 on success
        $demo = 1
    else
        ; return 0 on failure
        $demo = 0
    endif
endfunction
```

Step 2: Create a User Defined Script Configuration Rule

Include the file containing the custom function in your configuration by setting a configuration rule in the User Defined Scripts Configuration Module.

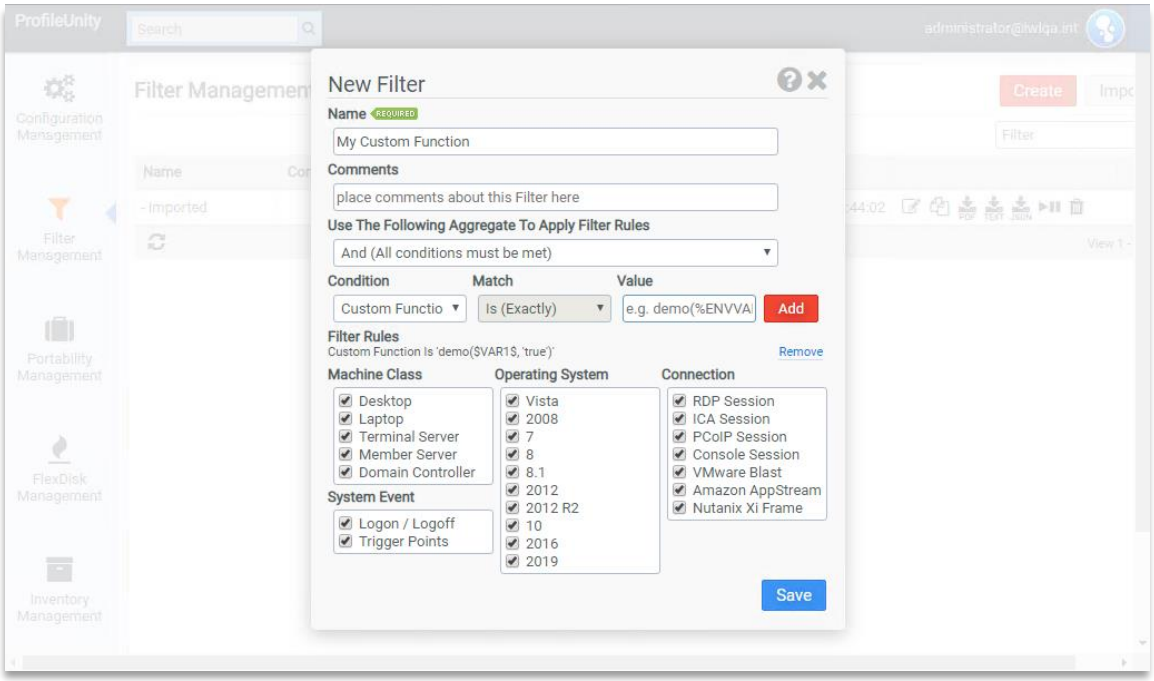


Example:

Execute: Before ProfileUnity at Logon
Type: KiXtart
File: @lserver\netlogon\my_function.kix
Run In-Process: Selected

Step 3: Create a Filter

Build a filter that uses the custom function.



Example:

Condition: Custom Function
Match: Is (Exactly)
Value: demo(%ENVVAR%, "some value")

Appendix E - KiXtart: Do You Care?

The following is an excerpt from the KiXtart help manual written by Mr. Ruud van Velsen.

Introduction

KiXtart was started in 1991 as a spare time project in response to the many requests for logon scripting functionality for the Microsoft LAN Manager environment. KiXtart's simplicity, speed and lack of competition soon made it very popular with LAN Manager network administrators.

KiXtart was initially distributed as freeware through bulletin boards in Europe. Later, Internet sites picked up on KiXtart and started distribution lists, discussion forums and script archives. KiXtart was also shipped as part of several Microsoft Resource Kits. Over time, KiXtart grew, both in popularity as well as in functionality. Windows NT and Windows 95 support was added, as well as lots of new functions and features.

Today, thousands of organizations worldwide use KiXtart. Banks, insurance companies, colleges, universities, hospitals, power plants, governmental organizations, IT companies, car manufacturers, oil companies, aerospace industries, publishers, amusement parks, broadcasting companies, and numerous other types of organizations around the globe make daily use of KiXtart to configure workstations, install software, and perform many other scripting tasks.

KiXtart has also become a hot topic on various Internet discussion forums, with many enthusiastic participants sharing tips, tricks and scripts.

Over the years, many people have asked when KiXtart would be commercialized. In fact, requests for pricing and licensing information on KiXtart are quite common.

If nothing else, all of this proves that KiXtart has a value.

Rather than commercializing KiXtart, I would like to turn its value into something truly positive. Specifically, I would like to use its value to help people who absolutely need and deserve our support: the people of Nepal.

As part of this initiative, KiXtart 2001 is provided to you as so-called CareWare. Exactly what this means is detailed in the following paragraphs. Please read the information carefully and support the KiXtart CareWare initiative!

What is CareWare?

CareWare is a variant on shareware and freeware. It is sometimes also known as 'charityware', 'donationware', 'helpware' or 'goodware', and is copyrighted software that you are allowed to use at no charge in return for a donation to specified charity/ies or to a charity of the users' choice.

KiXtart CareWare can be downloaded, installed and evaluated at no charge. If you continue using KiXtart, you are kindly requested to make a donation to a non-profit charitable organization. A list of preferred charities is provided below.

How much should we donate?

The answer to this question is in your heart. The donation amount should reflect your perception of the value of KiXtart for your organization. The suggested minimum donation amount is fifty US dollars ($50) per organization/company using KiXtart. Please consider that CareWare is not about making money, but about sharing with and caring for other people.

Making a donation is more important than the actual amount of the donation.

Note that in many countries, charitable donations to officially registered charities are tax deductible, so you may be able to donate more than you think!

Who should we donate to?

The following non-profit, charitable organizations that support the people in Nepal are preferred:

Room to Read

World change starts with educated children™

http://www.roomtoread.org/

Room to Read seeks to provide every child with an opportunity to gain the lifelong gift of literacy by attacking the root causes of illiteracy in Nepalese society.

A dedicated group of unpaid volunteers established the foundation in 1998. One village at a time, one school at a time, the Books for Nepal project is reaching out to communities to provide the gift of education.

Note: the Room-to-Read organization was formerly known as Books-for-Nepal.

ROKPA INTERNATIONAL

http://www.rokpa.org

ROKPA INTERNATIONAL is a non-profit organization helping and supporting people in need irrespective of their nationality, religion or cultural background.

ROKPA INTERNATIONAL works in the areas of education, health care, relief of hunger and preservation of culture, self-help and ecology. The organization both offers emergency and long-term help through its projects in Nepal, Tibet and other countries.

If, for whatever reason, you cannot donate to these particular organizations, you are kindly requested to donate to Unicef instead:

http://www.unicef.org

For more than 53 years UNICEF has been helping governments, communities and families make the world a better place for children. Part of the United Nations system, UNICEF has an enviable mandate and mission, to advocate for children's rights and help meet their needs.

Note: more details on these organizations can be found in the GuideStar directory: http://www.guidestar.org.

Why Nepal?

When I visited Nepal in 1999, I became enchanted with its magnificent beauty and its kind and hospitable people. At the same time, I was stunned by the poverty.

Nepal, home of Mount Everest, is one of the poorest countries in the world in relative as well as absolute terms. More than half of the population lives below the poverty line and 53% of the people live on less than US$ 1 per day. Nepal has few natural resources apart from its beauty and hardworking people. Life

expectancy is very low, and illiteracy affects more than 50% of the children. Education, medication, and even basic things such as clean water are a luxury in large parts of Nepal. Malnutrition is another widespread problem: everyday, a Nepali child goes blind for want of vitamin A, something that can be prevented by a medicine costing less than ten cents.

What do I get in return?

Of course, the whole concept of CareWare is about giving, not receiving. However, making a donation on behalf of KiXtart provides the following benefits:

- People elsewhere in the world benefit from your support.
- You get to feel good about using KiXtart.
- You motivate me to continue developing KiXtart.

Additionally, if you choose to register your donation, you will be kept up to date on KiXtart developments, and your (company) name can be included on the list of KiXtart CareWare sponsors. See below for details on how to register your donation.

How should we make a donation?

To make a donation, simply select the organization you would like to support, determine the amount you can donate, and use one of the donation methods supported by the organization.

When you make a donation, please include a reference to "KiXtart 2001".

Optionally, you can also register your donation by forwarding the confirmation email you send to or receive from the charitable organization to kixtart2001@hotmail.com or ruudv@microsoft.com.

I can't make a donation to charity!

If you are not able to donate money to any charity, for whatever reason, I would appreciate it if you could let me know why. Understanding what the problem with making a donation is will enable me to improve the KiXtart CareWare process.

I don't care…

That is entirely your prerogative. The KiXtart CareWare initiative is based on your voluntary cooperation. KiXtart has no built-in registration process or license checks.

Please carefully consider the value of KiXtart to you and your organization, and reconsider making a donation. Your support will be greatly appreciated, by me, and more importantly, by the organizations you donate to and the people they support. Join the growing number of KiXtart CareWare supporters today!