

SmartHub Console User Guide

March 31, 2016

1.1 Edition

SmartHub Console User Guide 1.1

Copyright © 2016 Citrix Systems, Inc. All Rights Reserved.

Version 1.1

Citrix Systems, Inc.

851 West Cypress Creek Road

Fort Lauderdale, FL 33309

United States of America

Disclaimers

This document is furnished "AS IS." Citrix, Inc. disclaims all warranties regarding the contents of this document, including, but not limited to, implied warranties of merchantability and fitness for any particular purpose. This document may contain technical or other inaccuracies or typographical errors. Citrix, Inc. reserves the right to revise the information in this document at any time without notice. This document and the software described in this document constitute confidential information of Citrix, Inc. and its licensors, and are furnished under a license from Citrix, Inc.

Citrix Systems, Inc., the Citrix logo, Citrix XenServer and Citrix XenCenter, are trademarks of Citrix Systems, Inc. and/or one or more of its subsidiaries, and may be registered in the United States Patent and Trademark Office and in other countries. All other trademarks and registered trademarks are property of their respective owners.

Trademarks

Citrix®
XenServer®
XenCenter®
XenDesktop®
XenApp®
Podio®

Table of Contents

Welcome	4
Feature Summary.....	4
Feature Detail	5
Alerts	5
Add a Notification	5
Concierge	7
AutoLaunch	8
Add an AutoLaunch Item	8
Application Request	9
Application Request Podio Integration	9
Tag Editor	15
Add a Tag.....	15
Tag Examples.....	16
System Information	17
Add a System Info Object.....	17
System Info Object Examples.....	18

Welcome

This guide steps through the implementation, configuration and use of SmartHub.

Feature Summary

- Alerts
 - *Alert or notify users*
 - *Smart tag support*
 - *HTML format*
- Concierge
 - *GoToAssist live chat support*
 - *Supports GTA keyboard and mouse control*
- Auto Launch
 - *Enable or disable Windows startup applications*
- Application Request
 - *Request applications in an easy to use interface*
 - *Integrated with Podio*
- Tags
 - *PowerShell based*
 - *Build custom tags based on any information on the client*

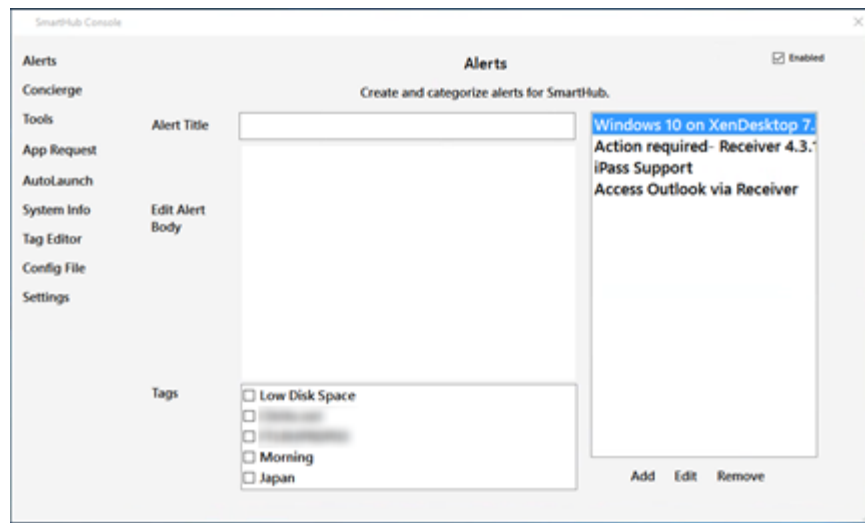
Feature Detail

Alerts

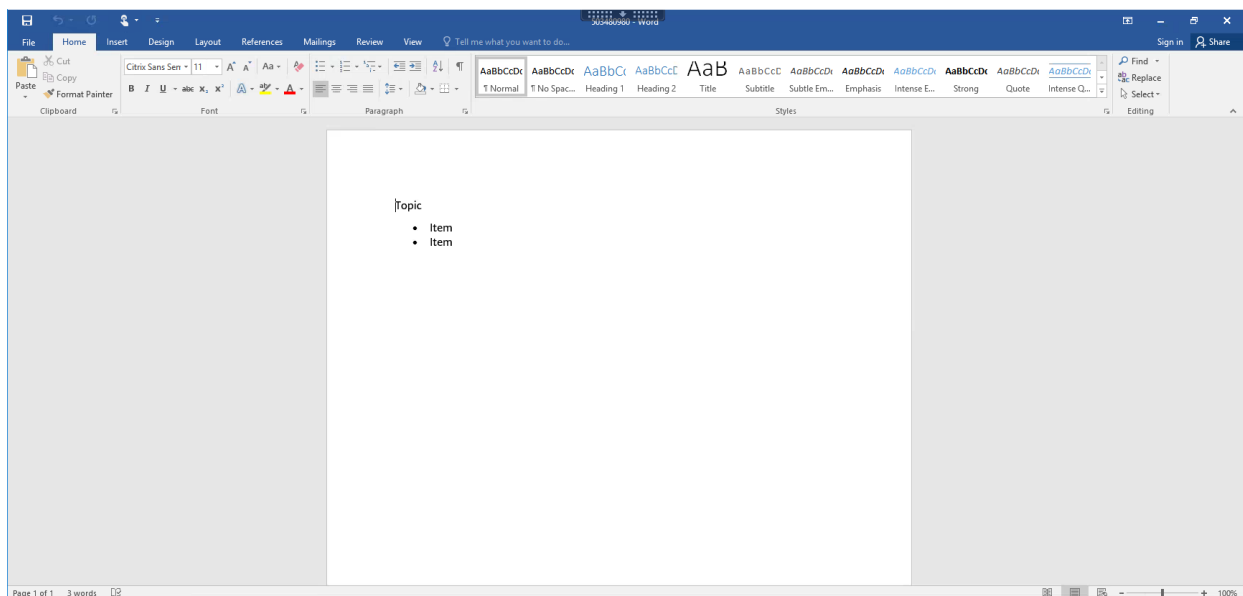
Alert many users or few. Fully customizable alerts can be sent to any user of SmartHub based on any criteria on their endpoint. Informed end users call support less and are much more productive.

Add a Notification

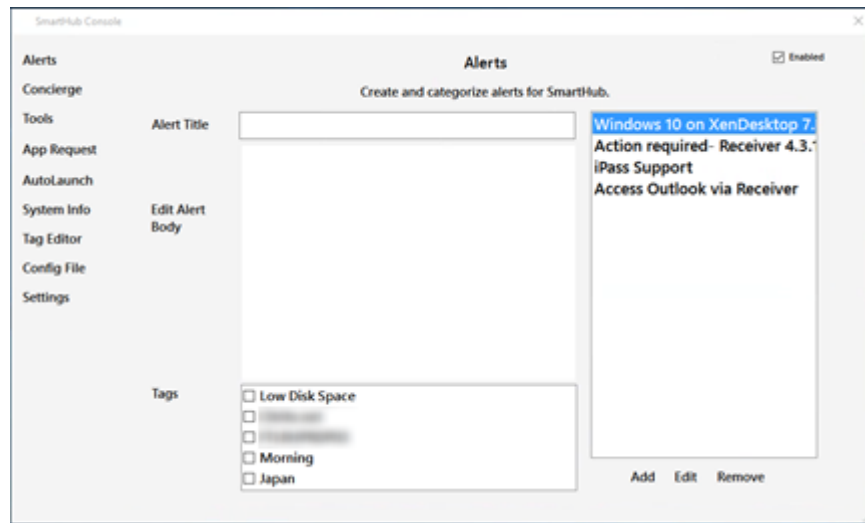
1. Launch the SmartHub Console and click on Alerts.



2. Add a title for the alert and click on “Edit Alert Body.” Edit this document directly or copy HTML data to the document.



3. Check a tag that pertains to the alert.



The image shows a screenshot of the 'SmartHub Console' window, specifically the 'Alerts' section. The window has a title bar 'SmartHub Console' and a close button. On the left is a sidebar with a list of navigation items: Alerts, Concierge, Tools, App Request, AutoLaunch, System Info, Tag Editor, Config File, and Settings. The main area is titled 'Alerts' and contains the instruction 'Create and categorize alerts for SmartHub.' Below this, there are two input fields: 'Alert Title' and 'Edit Alert Body'. To the right of these fields is a preview area showing a sample alert: 'Windows 10 on XenDesktop 7.' followed by 'Action required- Receiver 4.3.', 'iPass Support', and 'Access Outlook via Receiver'. Below the input fields is a 'Tags' section with a list of checkboxes: 'Low Disk Space', 'Morning', and 'Japan'. At the bottom right of the main area are three buttons: 'Add', 'Edit', and 'Remove'. The 'Alerts' section is also marked as 'Enabled' in the top right corner.

SmartHub Console

Alerts

Enabled

Create and categorize alerts for SmartHub.

Alert Title

Edit Alert Body

Tags

- ☐ Low Disk Space
- ☐ Morning
- ☐ Japan

Windows 10 on XenDesktop 7.
Action required- Receiver 4.3.
iPass Support
Access Outlook via Receiver

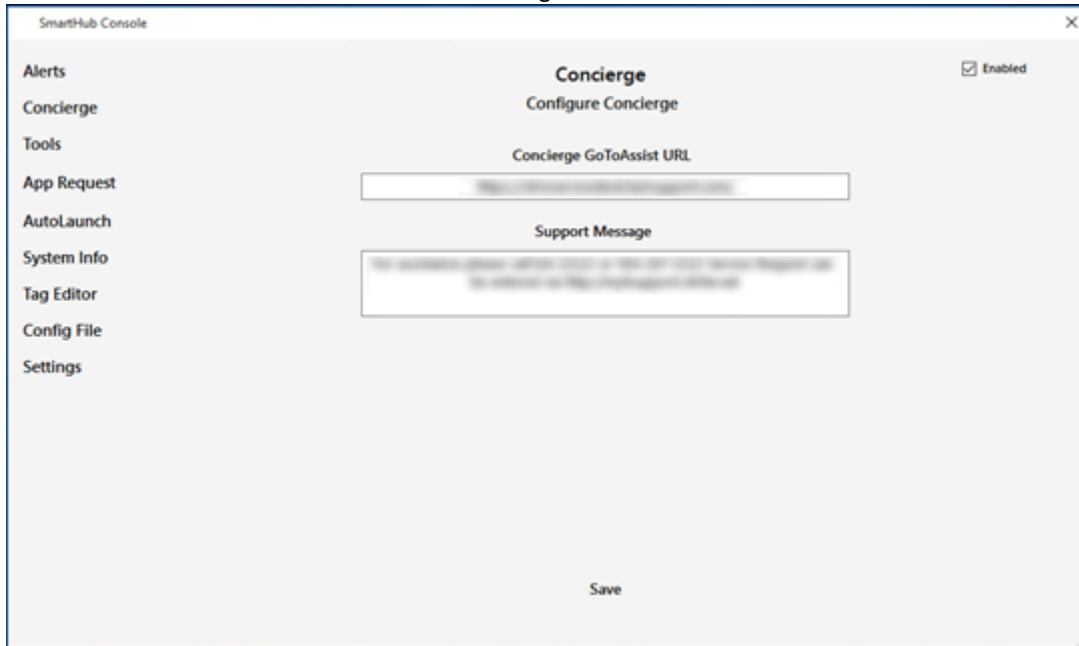
Add Edit Remove

4. Click add.

Concierge

GoToAssist live chat support.

1. Launch the SmartHub Console. Click on Concierge.



The screenshot shows the 'SmartHub Console' window with a sidebar on the left containing the following menu items: Alerts, Concierge, Tools, App Request, AutoLaunch, System Info, Tag Editor, Config File, and Settings. The 'Concierge' menu item is selected. The main content area is titled 'Concierge' and 'Configure Concierge'. In the top right corner of this area is a checkbox labeled 'Enabled' which is checked. Below the title, there is a section for 'Concierge GoToAssist URL' with a text input field containing a placeholder URL. Underneath that is a section for 'Support Message' with a larger text input field containing a placeholder message. At the bottom center of the console window is a 'Save' button.

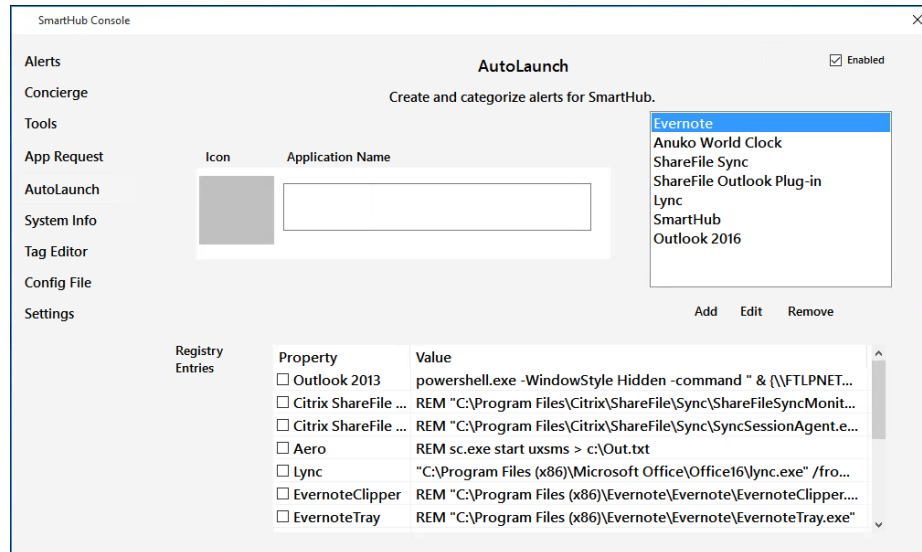
2. Edit the GoToAssist URL and Support Message. Click Save.

AutoLaunch

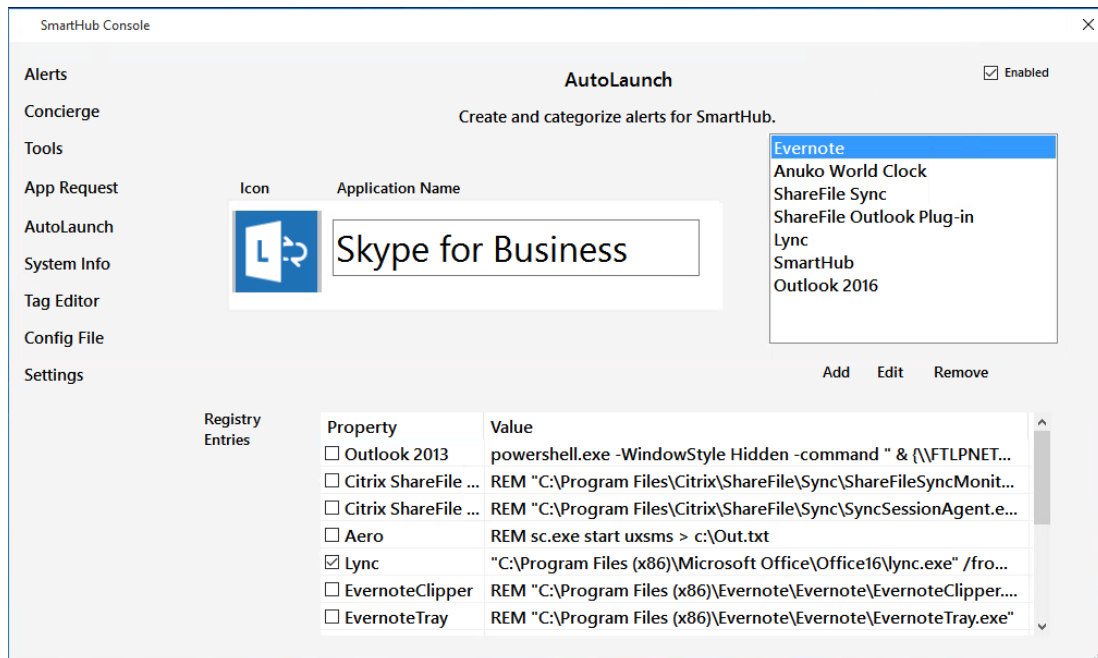
Enable or disable Windows startup applications. Start-Up applications can enable a more productive user experience or slow down logon speeds. This feature of SmartHub allows the user chose their own start-up experience.

Add an AutoLaunch Item

1. Start the SmartHub Console. Click on AutoLaunch



2. Click on the Gray box to select an icon. Add an application name.

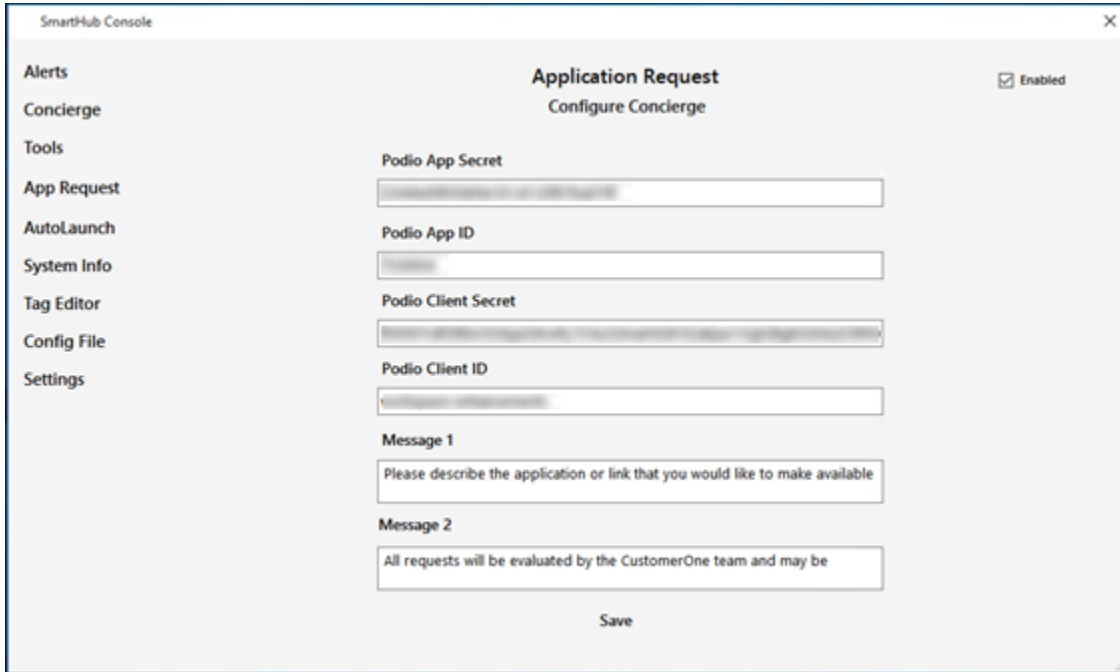


3. Check the registry entries that enable the application to launch on startup. Click Add.

Application Request

Request applications with an easy to use interface.

1. Launch the SmartHub Console, click on App Request.



The screenshot shows the 'SmartHub Console' window with a sidebar on the left containing links: Alerts, Concierge, Tools, App Request (highlighted), AutoLaunch, System Info, Tag Editor, Config File, and Settings. The main area is titled 'Application Request' with a subtitle 'Configure Concierge' and an 'Enabled' checkbox. The form contains several input fields: 'Podio App Secret', 'Podio App ID', 'Podio Client Secret', and 'Podio Client ID'. Below these are two message boxes: 'Message 1' with the placeholder text 'Please describe the application or link that you would like to make available' and 'Message 2' with the text 'All requests will be evaluated by the CustomerOne team and may be'. A 'Save' button is at the bottom.

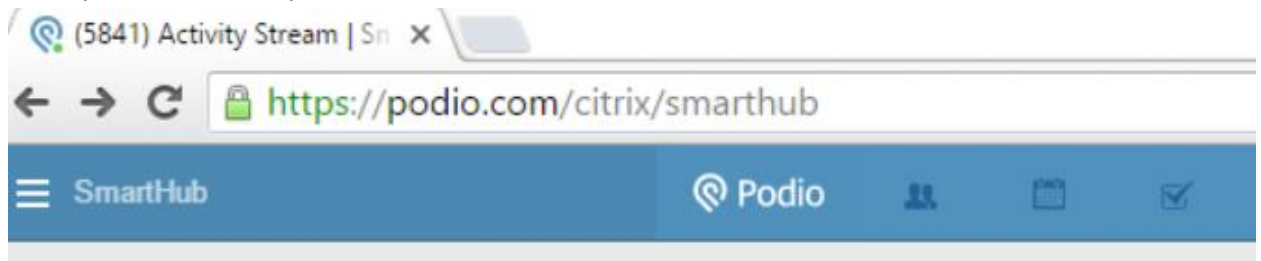
2. Fill in the Podio application authentication information. Click on Save.

Application Request Podio Integration

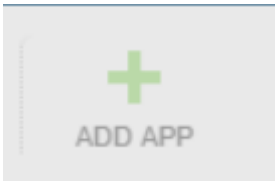
SmartHub integrates with the SmartHub Application Request App on Podio. This will guide you on adding the SmartHub Request App on your Podio workspace and the details needed to integrate with the SmartHub console.

All application fields can be customized per your organization's preferences except for the following.

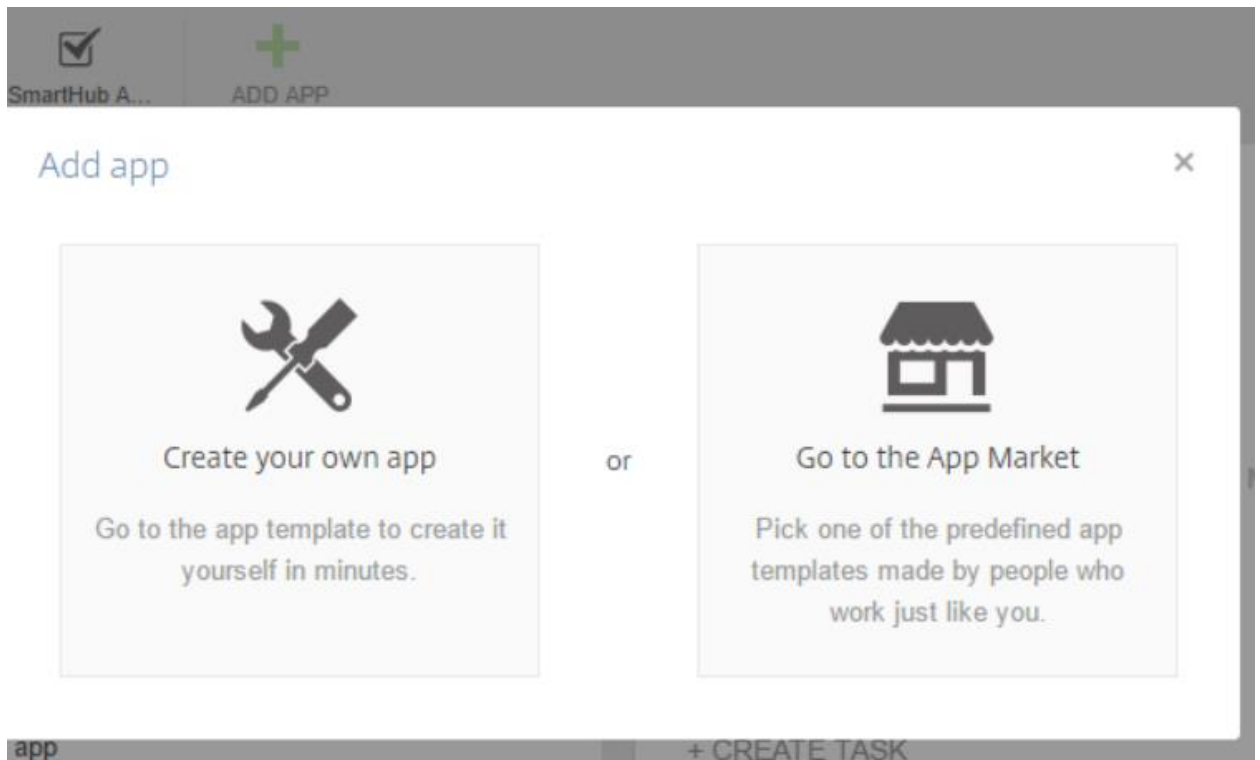
- External ID requested-by – Field ID 118691331
 - External ID additional-ie-requirements-and-components -- Field ID 118691336
1. Go to your Podio workspace or create a new one for free at www.Podio.com.



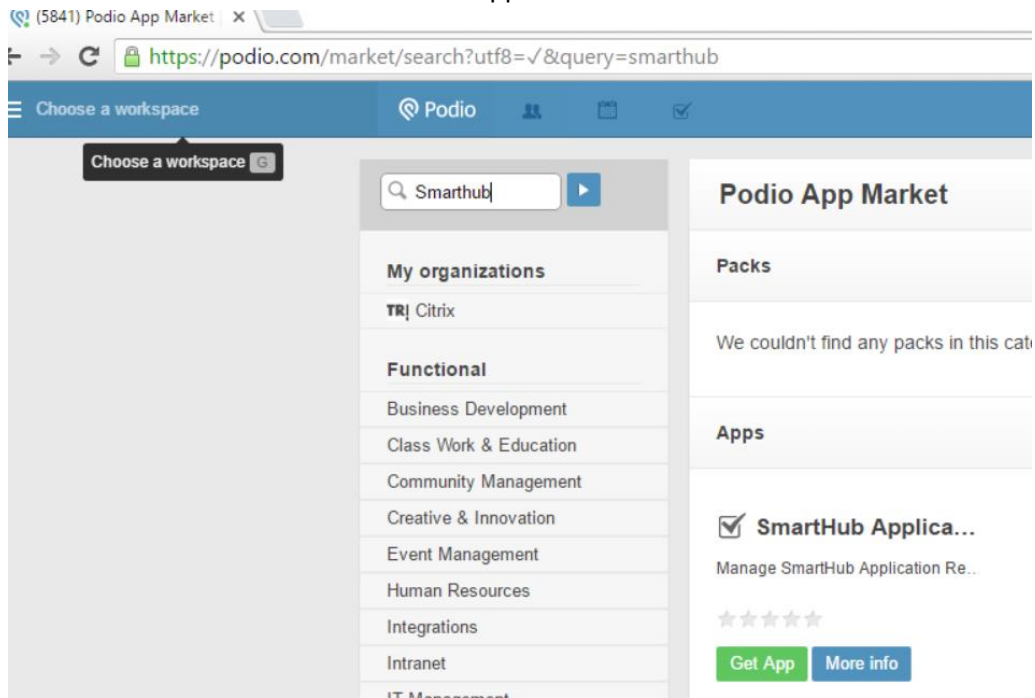
2. Click on Add App



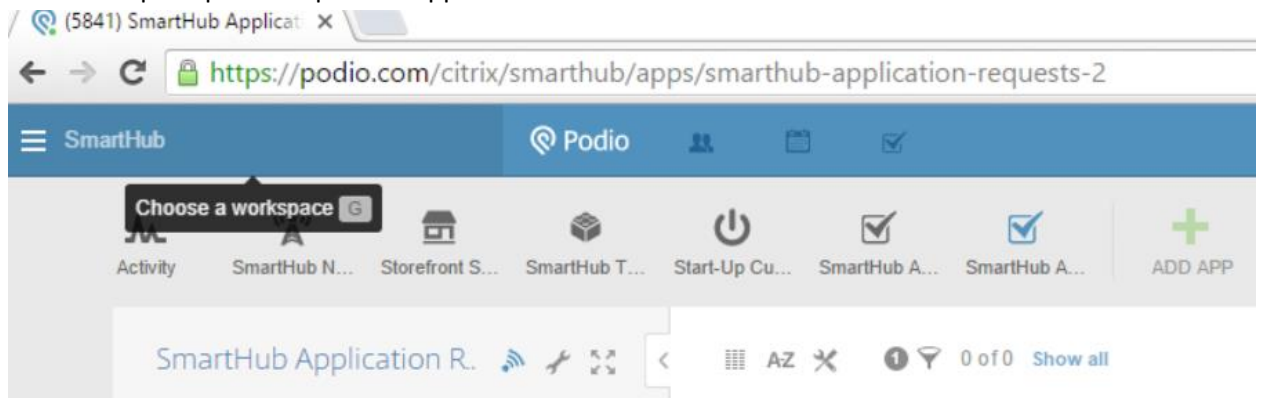
3. Click on “Go to the App Market”



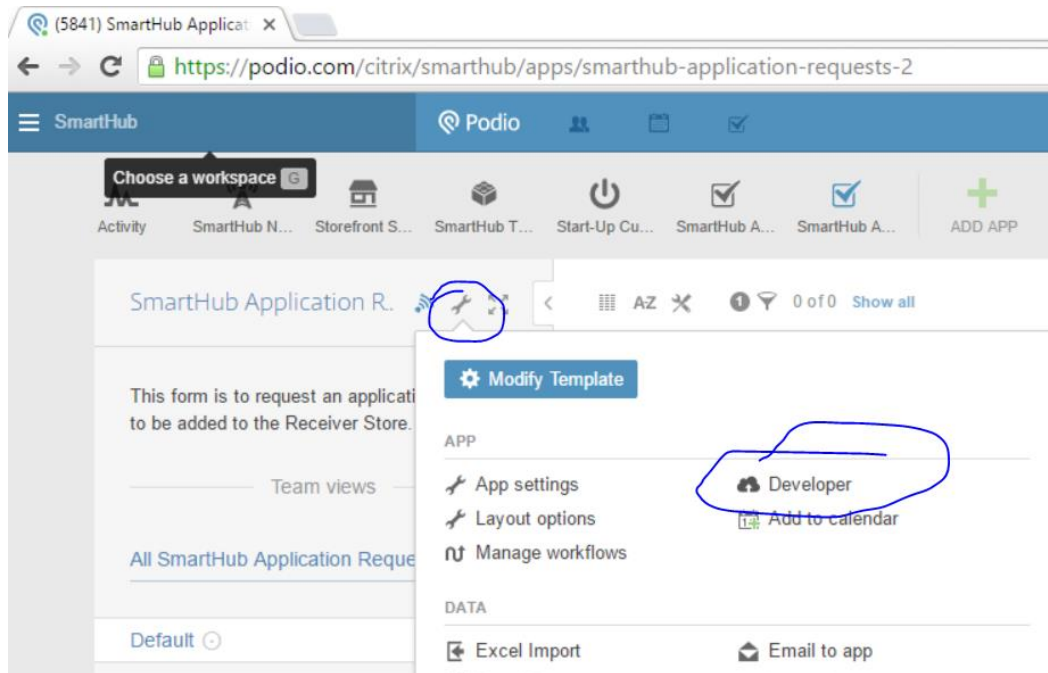
4. Search for SmartHub and click on “Get App”



5. Follow the prompts and open the app in Podio.



6. Click on the wrench and select “Developer”



7. Copy the App ID and App Token\Secret – these values must be added to the SmartHub Console – the data will not match this example!

Podio:

Developer

App token

You can use the app token to authenticate as an app instead of a user. When you authenticate as an app you can only access data from that app and documentation for details.

App ID for SmartHub Application Requests:

Token for SmartHub Application Requests:

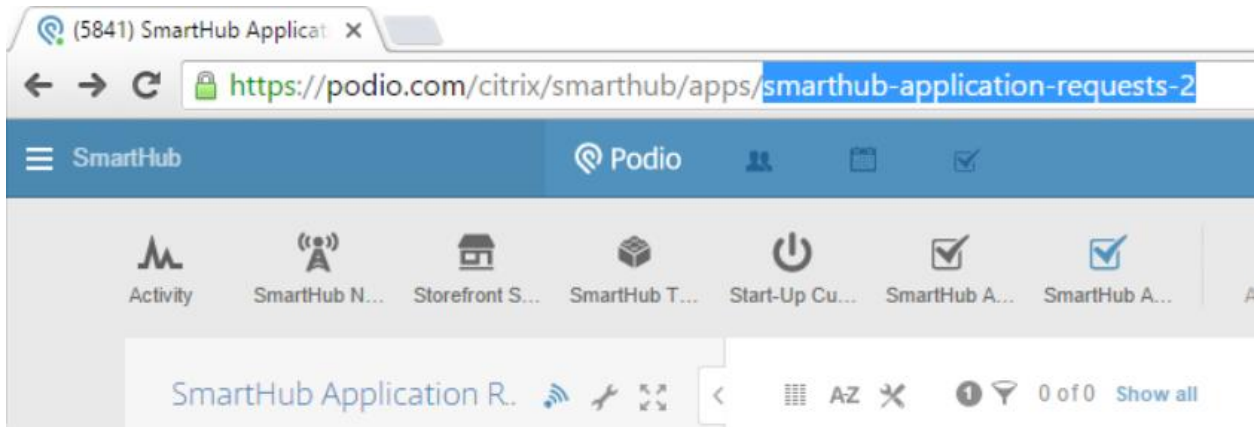
Regenerate App Token

SmartHub Console – App Requests:

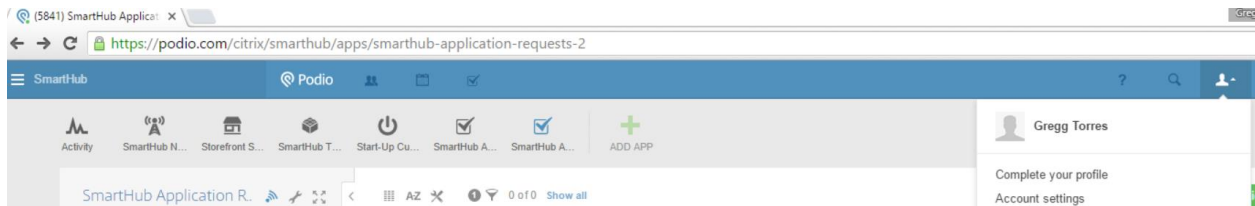
Podio App Secret

Podio App ID

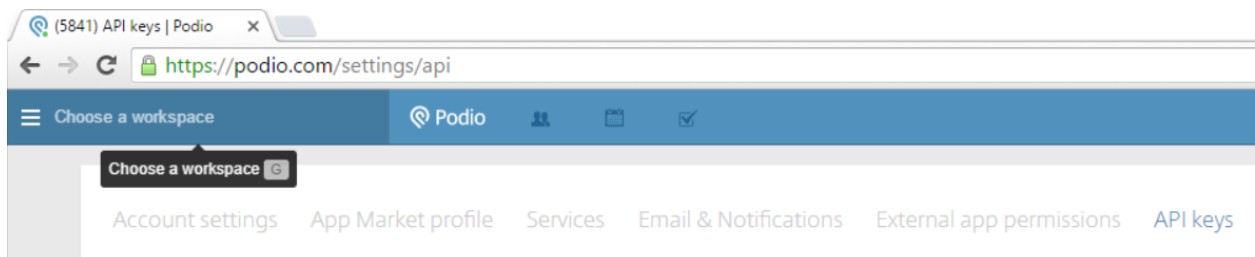
8. Copy the application name – shown here



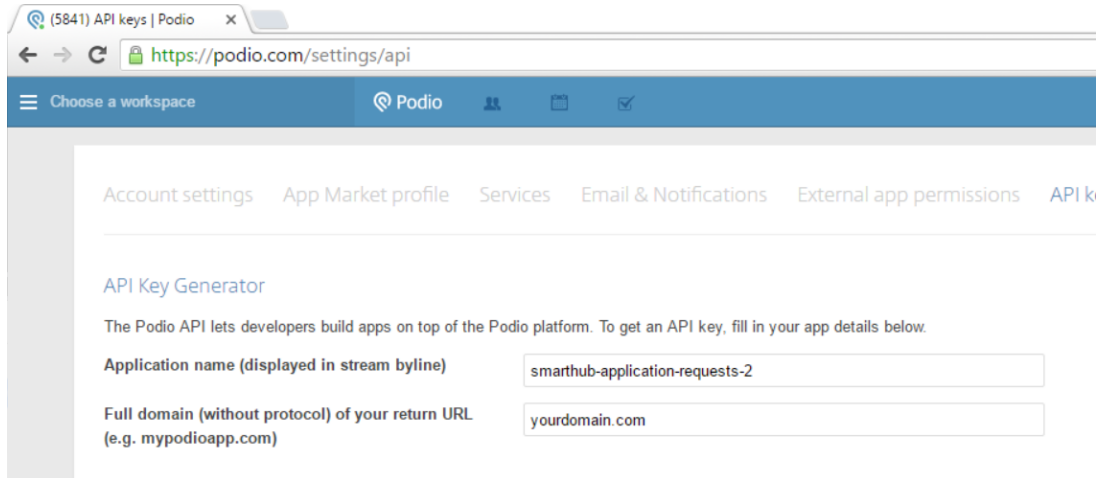
9. Click on the user icon on the application screen, select “Account Settings”
a. User must have full rights to the Podio workspace.



10. Click on “API Keys”



11. Type in the application name saved from step number 8, your domain name and click on “Generate API Key”



The screenshot shows a web browser window with the address bar displaying `https://podio.com/settings/api`. The page title is "(5841) API keys | Podio". The navigation bar includes a menu icon, "Choose a workspace", the Podio logo, and icons for users, a calendar, and a checkmark. The main content area has a breadcrumb trail: "Account settings" > "App Market profile" > "Services" > "Email & Notifications" > "External app permissions" > "API k". Below this is the "API Key Generator" section. It contains a paragraph: "The Podio API lets developers build apps on top of the Podio platform. To get an API key, fill in your app details below." There are two input fields: "Application name (displayed in stream byline)" with the value "smarthub-application-requests-2", and "Full domain (without protocol) of your return URL (e.g. mypodioapp.com)" with the value "yourdomain.com".

12. Copy the client ID and Client Secret and put the values in the SmartHub Console.



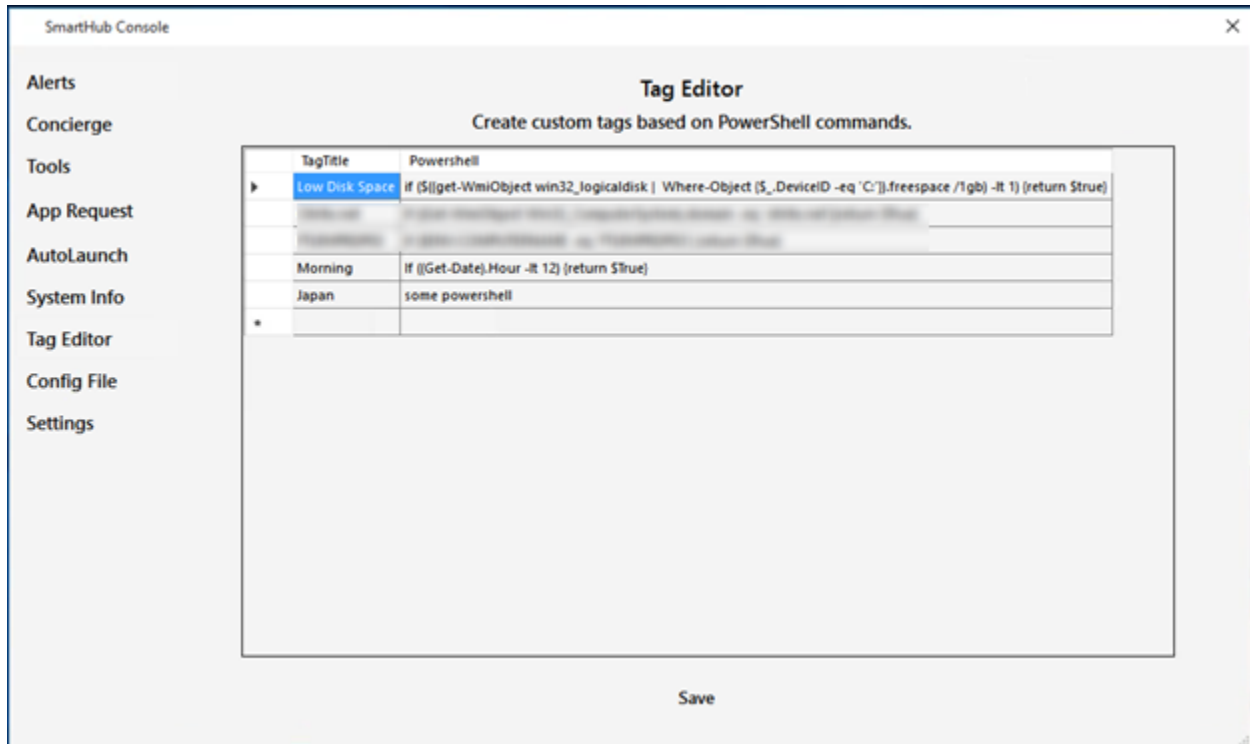
The screenshot shows two input fields on a light gray background. The first field is labeled "Podio Client Secret" and contains a long, complex alphanumeric string. The second field is labeled "Podio Client ID" and contains a shorter alphanumeric string.

Tag Editor

Custom tagging support based on Windows PowerShell. Alerts can be based off of almost anything.

Add a Tag

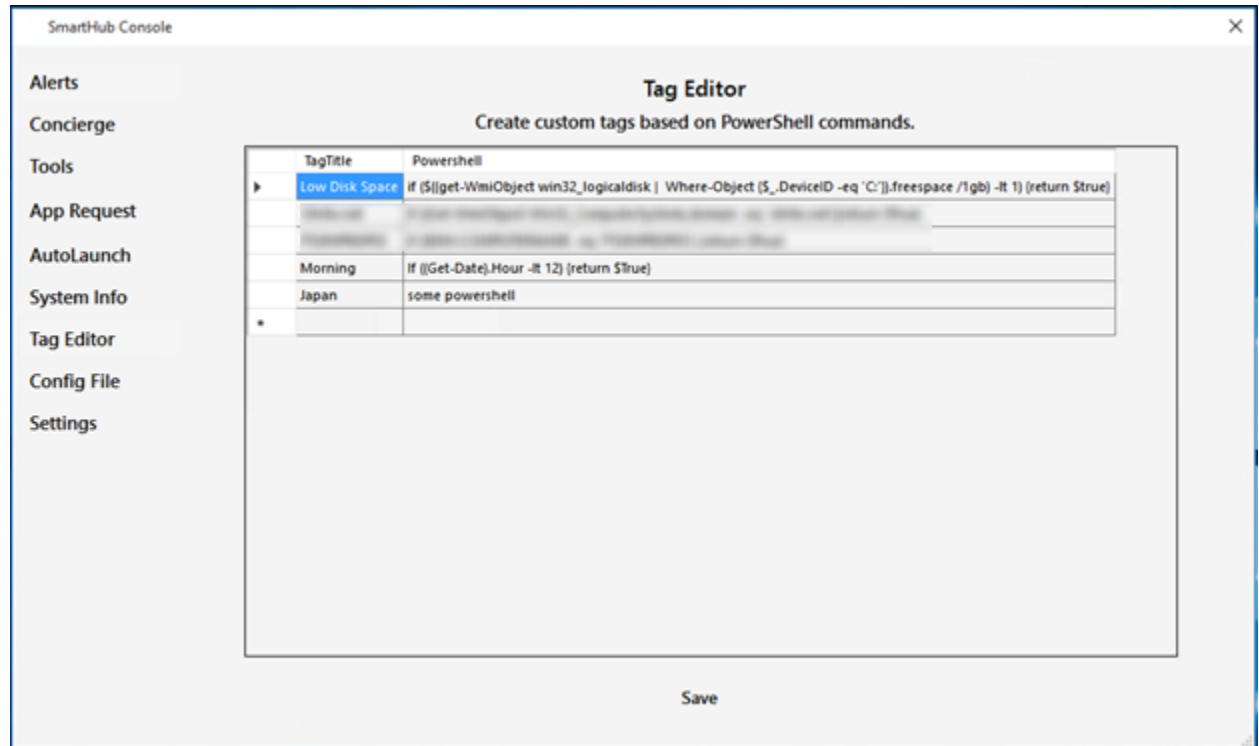
1. Launch the SmartHub Console. Click on Tag Editor.



2. Type your tag title. Add a powershell script that returns a \$true. For example; we would like to target our Japanese users. Their desktop names start with "JPN." The following powershell will return a \$true if that is the case.

```
if ($ENV:COMPUTERNAME -match "JPN") {Return $True}
```

Copy this statement to the powershell column.



3. Click Save. This tag can now be used to target alerts.

Tag Examples

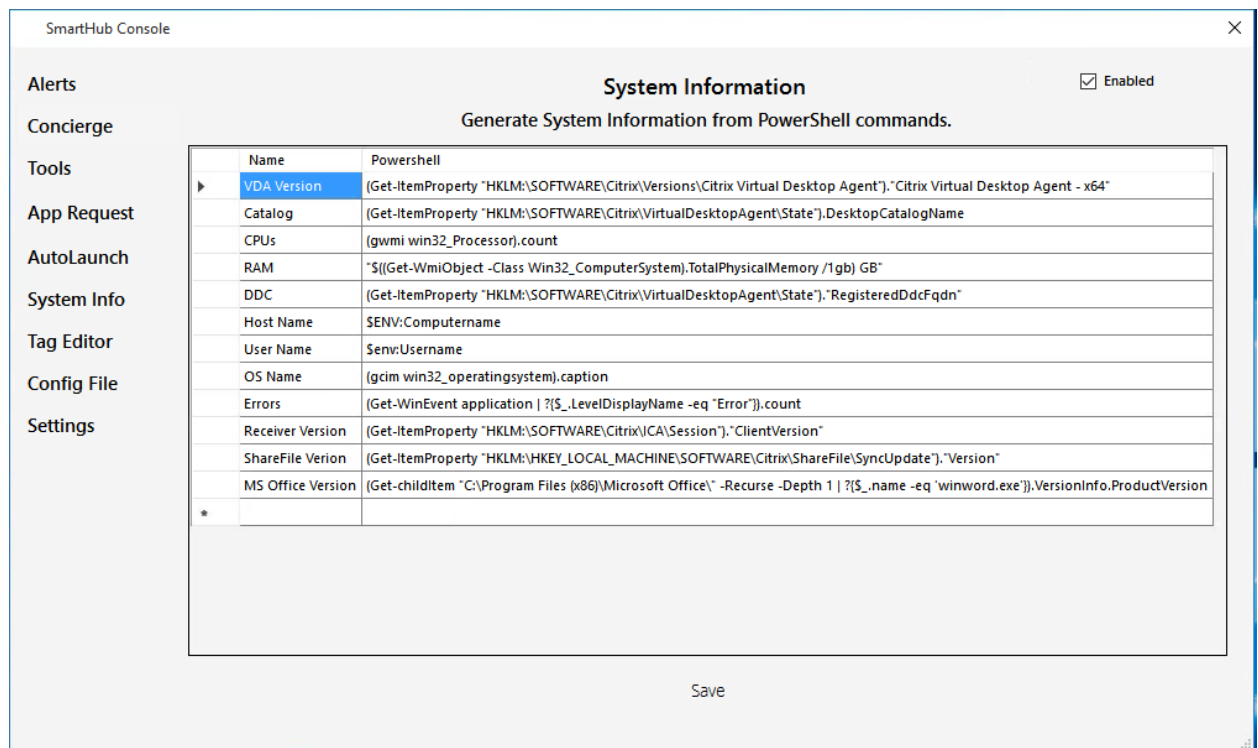
Low Disk Space	if ((\$get-WmiObject win32_logicaldisk Where-Object {\$_.DeviceID -eq 'C:'}).freespace /1gb) -lt 1) {return \$true}
On Domain	if ((Get-WmiObject Win32_ComputerSystem).domain -eq yourdomain.com'){return \$True}
Morning	If ((Get-Date).Hour -lt 12) {return \$True}
Japan	if (\$ENV:COMPUTERNAME -match "JPN") {return \$true}
All Users	return \$true

System Information

Custom system information gathered via Windows PowerShell. Data can be based off of any local PowerShell command.

Add a System Info Object

1. Launch the SmartHub Console. Click on System Info.



2. Add a name and a PowerShell command. Click Save. The text from the PowerShell command will be displayed on the SmartHub client.

System Info Object Examples

VDA Version	(Get-ItemProperty "HKLM:\SOFTWARE\Citrix\Versions\Citrix Virtual Desktop Agent")."Citrix Virtual Desktop Agent - x64"
Catalog	(Get-ItemProperty "HKLM:\SOFTWARE\Citrix\VirtualDesktopAgent\State").DesktopCatalogName
CPUs	(gwmi win32_Processor).count
RAM	\$((Get-WmiObject -Class Win32_ComputerSystem).TotalPhysicalMemory /1gb) GB
DDC	(Get-ItemProperty "HKLM:\SOFTWARE\Citrix\VirtualDesktopAgent\State")."RegisteredDdcFqdn"
Host Name	\$ENV:Computername
User Name	\$env:Username
OS Name	(gcim win32_operatingsystem).caption
Errors	(Get-WinEvent application ?{\$_.LevelDisplayName -eq "Error"}).count
Receiver Version	(Get-ItemProperty "HKLM:\SOFTWARE\Citrix\ICA\Session")."ClientVersion"
ShareFile Verion	(Get-ItemProperty "HKLM:\HKEY_LOCAL_MACHINE\SOFTWARE\Citrix\ShareFile\SyncUpdate")."Version"
MS Office Version	(Get-childitem "C:\Program Files (x86)\Microsoft Office\" -Recurse -Depth 1 ?{\$_.name -eq 'winword.exe'}).VersionInfo.ProductVersion