

Publishing, Installing and Upgrading Add-ins



Agenda

- Creating the App Catalog
- Publishing Apps in the App Catalog
- Installing and Upgrading Apps
- Deploying Provider-hosted Apps



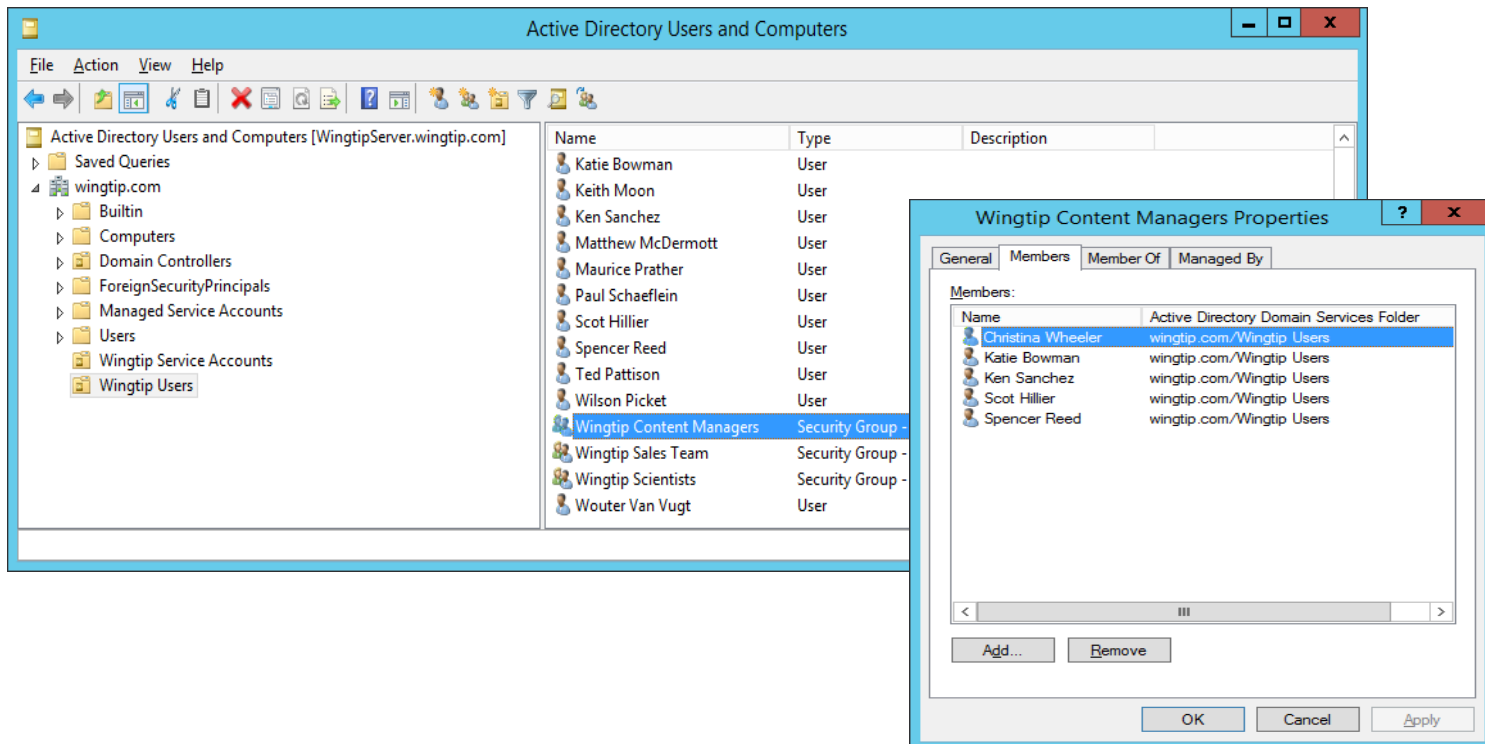
Understanding the App Catalog

- App publishing scheme based on App Catalog
 - App Catalog is site collection with special doc library
 - App packages are published (uploaded) to app catalog
 - Provides better app discovery, installation and upgrade
- App Catalog in on-premises farms
 - One App Catalog site required for each web application
 - End users often play role of App Catalog administrator
- App Catalog in Office 365 & SharePoint Online
 - One App Catalog site used to manage entire tenancy



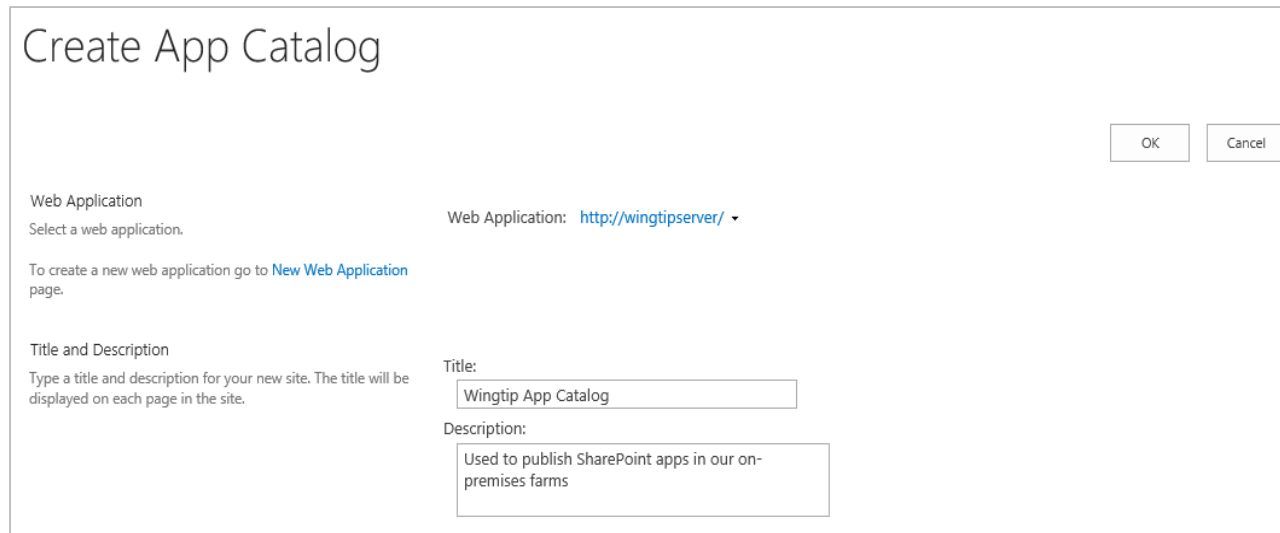
Who Manages the App Catalog?

- App Catalog can be managed by business users
 - IT staff doesn't have to install and manage apps
 - Security can be configured using Active Directory group



Creating the App Catalog Site Collection

- You must create the App Catalog site collection
 - You can create it using a PowerShell script
 - You can create it using Central Administration
 - App Catalog site associated with one web application



The screenshot shows the 'Create App Catalog' dialog box. It has a title bar 'Create App Catalog' and 'OK' and 'Cancel' buttons in the top right corner. The dialog is divided into two main sections. The top section is titled 'Web Application' and contains the text 'Select a web application.' followed by 'Web Application: <http://wingtipserver/>'. Below this, it says 'To create a new web application go to [New Web Application](#) page.' The bottom section is titled 'Title and Description' and contains the text 'Type a title and description for your new site. The title will be displayed on each page in the site.' It has two input fields: 'Title:' with the value 'Wingtip App Catalog' and 'Description:' with the value 'Used to publish SharePoint apps in our on-premises farms'.

Create App Catalog

OK Cancel

Web Application

Select a web application.

Web Application: <http://wingtipserver/>

To create a new web application go to [New Web Application](#) page.

Title and Description

Type a title and description for your new site. The title will be displayed on each page in the site.

Title:

Wingtip App Catalog

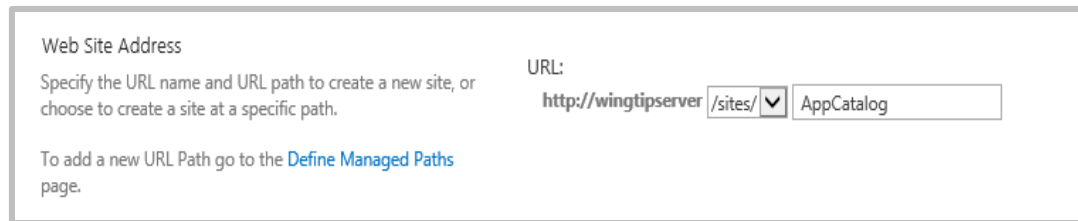
Description:

Used to publish SharePoint apps in our on-premises farms



App Catalog URL and Permissions

- App catalog site created at a specific URL
 - Creating App Catalog site with PowerShell is more flexible
you can create site as top-level domain using host-named site collections (HNSCs)



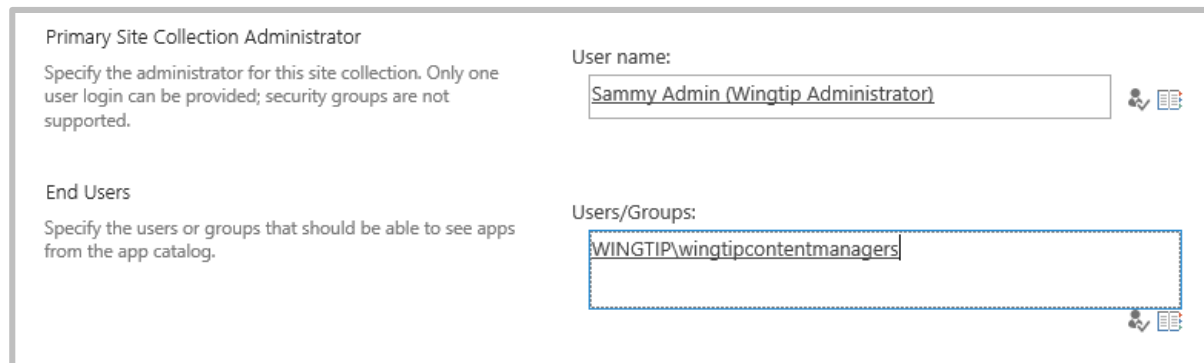
Web Site Address

Specify the URL name and URL path to create a new site, or choose to create a site at a specific path.

URL:

To add a new URL Path go to the [Define Managed Paths](#) page.

- Setting App Catalog permissions
 - Site collection administrator becomes App Catalog administrator
 - End user permissions allows user to discover and install apps



Primary Site Collection Administrator

Specify the administrator for this site collection. Only one user login can be provided; security groups are not supported.

User name:

End Users

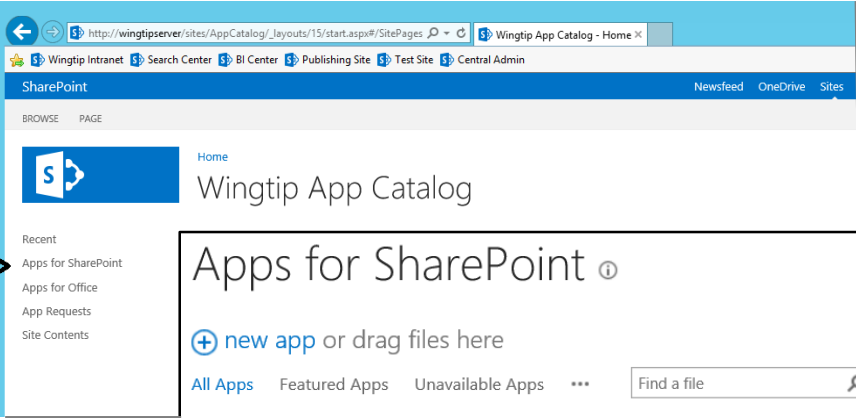
Specify the users or groups that should be able to see apps from the app catalog.

Users/Groups:



Apps for SharePoint Document Library

- Apps for SharePoint is special document library
 - It's the place where you publish SharePoint apps
 - You upload app package and enter the related metadata



The screenshot shows the 'Wingtip App Catalog' interface. On the left, a sidebar lists 'Recent' items: 'Apps for SharePoint', 'Apps for Office', 'App Requests', and 'Site Contents'. An arrow points from 'Apps for SharePoint' to the main content area. The main area is titled 'Apps for SharePoint' and includes a '+ new app or drag files here' button. Below this is a search bar and tabs for 'All Apps', 'Featured Apps', and 'Unavailable Apps'. A table lists installed apps with columns for Title, Name, App Version, Edit, Product ID, Metadata Language, and Default Metadata Language.

✓	📄	Title	Name	App Version	Edit	Product ID	Metadata Language	Default Metadata Language
Product ID : {921FDA40-576C-48CF-B00B-7285D24372A0} (1)								
✓	📄	Wingtip Search App	WingtipSearchApp	1.0.0.0	⋮	{921FDA40-576C-48CF-B00B-7285D24372A0}	English - 1033	Yes
Product ID : {2AB41D1E-C8A8-400A-AD6E-7C15FEE6A69A} (1)								
✓	📄	Calculator App	Calculator	2.0.0.0	⋮	{2AB41D1E-C8A8-400A-AD6E-7C15FEE6A69A}	English - 1033	Yes





DEMO

Creating an App Catalog Site

Agenda

- ✓ Creating the App Catalog
- Publishing Apps in the App Catalog
 - Installing and Upgrading Apps
 - Deploying Provider-hosted Apps



Add-in Deployment Overview

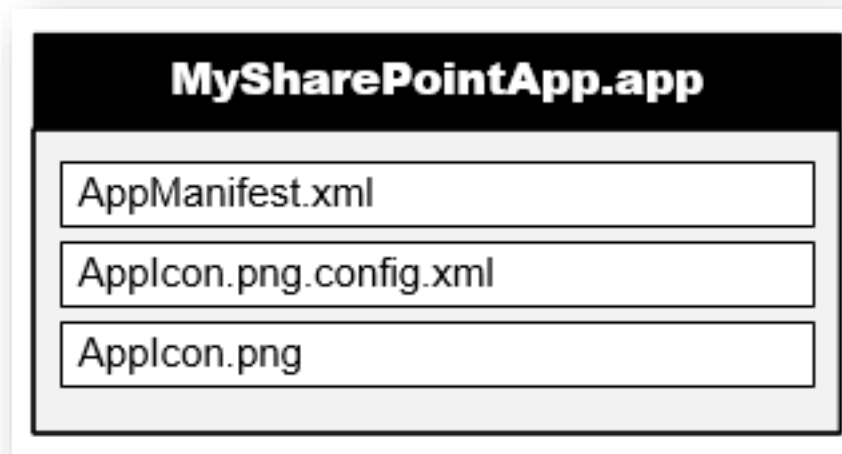
for a SharePoint-hosted Add-in

1. Package the add-in
 - Package add-in resources in distributable file
2. Publish the add-in
 - Make add-in available for installation and upgrade
3. Install the add-in
 - Make add-in available for use
4. Upgrade the add-in
 - Replace current version of add-in with newer version



App Package

- SharePoint apps distributed using app packages
 - App package is ZIP archive file with `*.app` extension
 - Built according to Open Package Convention (OPC)
 - Same packaging format used in Apps for Office
 - App package must contain `AppManifest.xml`
 - App package will often contain file for app icon



App Web Solution Package

- App package contains inner WSP for app web
 - Elements deployed to app web using solution package
 - Solution package built into app package as inner WSP



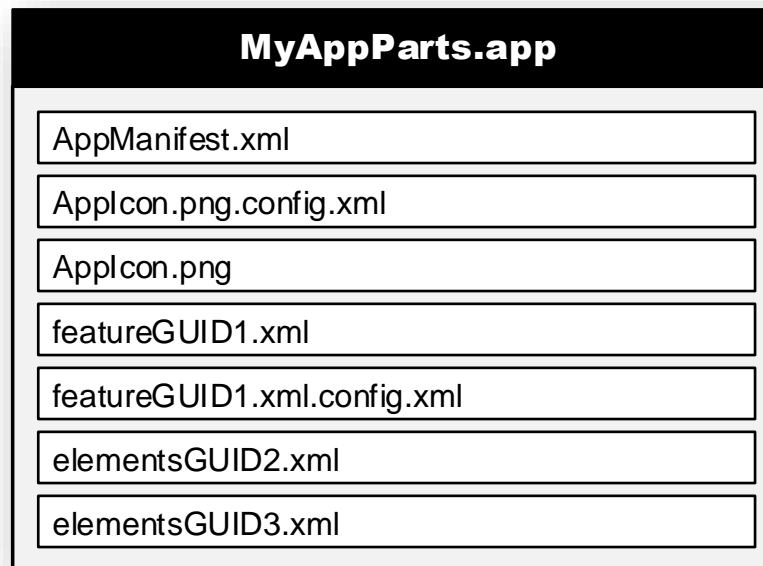
All SharePoint-hosted apps will have an inner WSP in their app package

Cloud-hosted apps will not have an inner WSP in their app package unless they have been implemented to create an app web



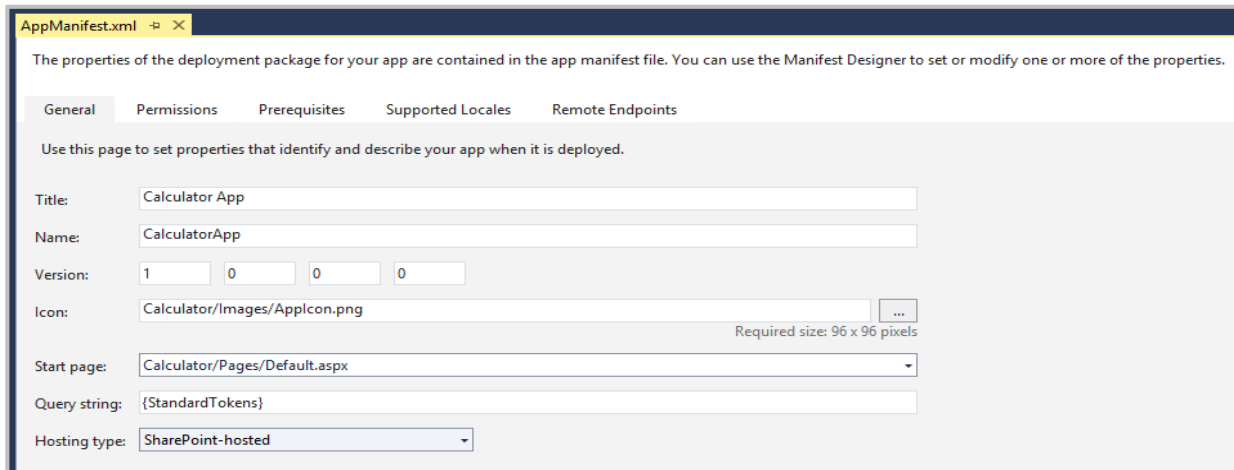
Packaging Host Web Features

- Host web feature elements added at top level
 - `elements.xml` file added for each app part
 - `elements.xml` file added for each UI custom action
 - `features.xml` file added for host web feature
- Visual Studio adds GUID to file names



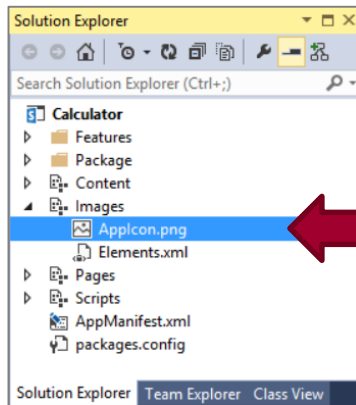
Preparing a Project for Packaging

- Review **AppManifest.xml** and make any required changes



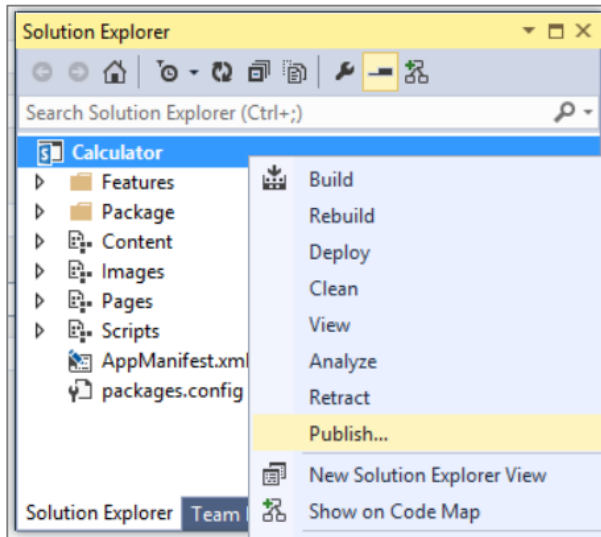
The screenshot shows the 'AppManifest.xml' Manifest Designer window. It has a title bar with 'AppManifest.xml' and standard window controls. Below the title bar is a descriptive text: 'The properties of the deployment package for your app are contained in the app manifest file. You can use the Manifest Designer to set or modify one or more of the properties.' Below this is a tabbed interface with tabs for 'General', 'Permissions', 'Prerequisites', 'Supported Locales', and 'Remote Endpoints'. The 'General' tab is selected. Below the tabs is a text area with the instruction: 'Use this page to set properties that identify and describe your app when it is deployed.' The form contains several fields: 'Title' (text box with 'Calculator App'), 'Name' (text box with 'CalculatorApp'), 'Version' (four text boxes for major, minor, build, and revision numbers, all containing '0'), 'Icon' (text box with 'Calculator/Images/AppIcon.png' and a browse button '...', with a note 'Required size: 96 x 96 pixels' below it), 'Start page' (dropdown menu with 'Calculator/Pages/Default.aspx' selected), 'Query string' (text box with '{StandardTokens}'), and 'Hosting type' (dropdown menu with 'SharePoint-hosted' selected).

- Add a custom App Icon

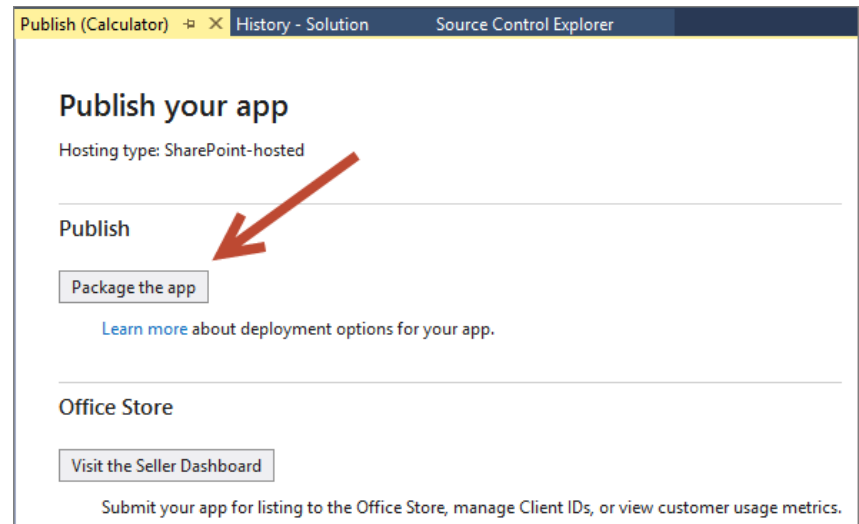


Creating an App Package

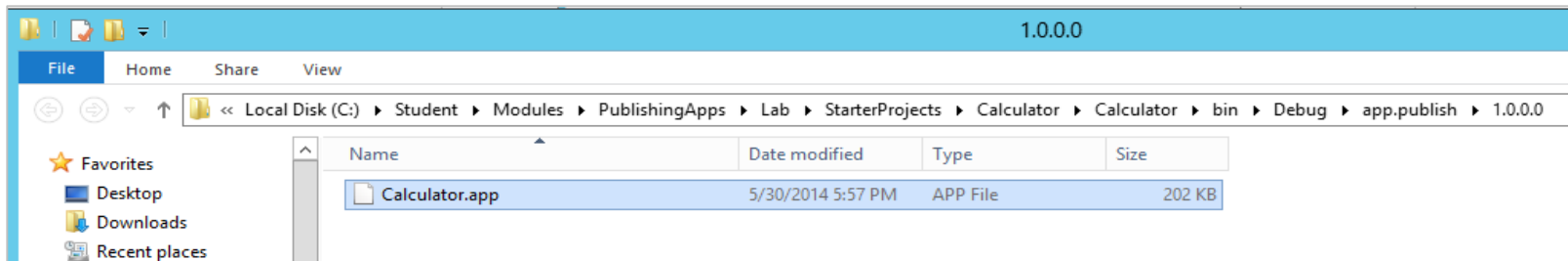
1



2

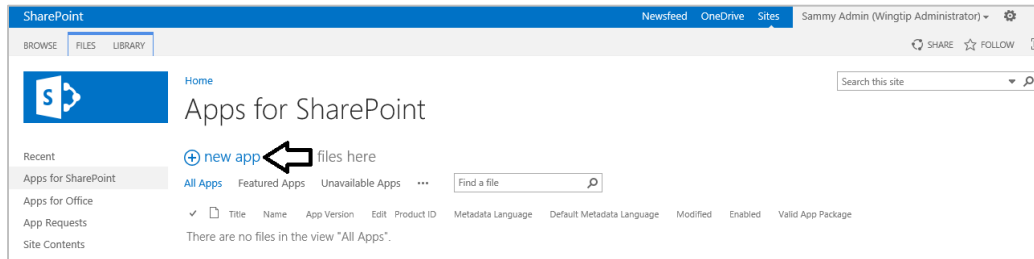


3



Publishing an App

- Upload app package to Apps for SharePoint library



- Enter app metadata to complete publishing process

A screenshot of the 'Apps for SharePoint - Calculator.app' metadata form. The form has a title bar 'Apps for SharePoint - Calculator.app'. Below the title bar is a ribbon with 'EDIT' and 'Commit' tabs. The 'EDIT' tab is active, showing icons for 'Save', 'Cancel', 'Paste', 'Copy', and 'Delete Item'. A yellow message box says: 'The document was uploaded successfully. Use this form to update the properties of the document.' The form fields are: 'Name' (Calculator.app), 'Title' (Calculator App), 'Metadata Language' (English), 'Default Metadata Language' (checked), 'Short Description' (Calculator App), and 'Description' (empty text area).A screenshot of the 'Featured' app settings section. It has a title bar 'Apps for SharePoint - Calculator.app'. The 'Enabled' checkbox is checked. The 'Featured' checkbox is also checked, with a yellow arrow pointing to it. Below the 'Featured' checkbox, there's a text box for 'Version: 1.0', and two lines of text: 'Created at 5/30/2014 5:47 PM by Sammy Admin (Wingtip Administrator)' and 'Last modified at 5/30/2014 5:47 PM by Sammy Admin (Wingtip Administrator)'. There are 'Save' and 'Cancel' buttons at the bottom right.

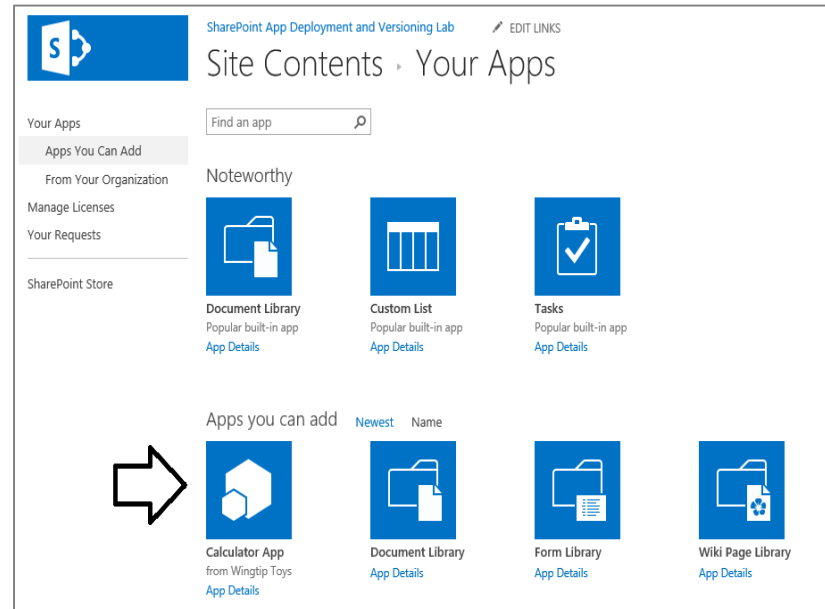
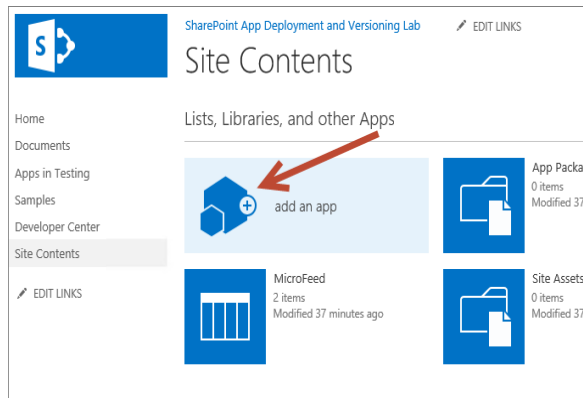
Agenda

- ✓ Creating the App Catalog
- ✓ Publishing Apps in the App Catalog
- Installing and Upgrading Apps
 - Deploying Provider-hosted Apps



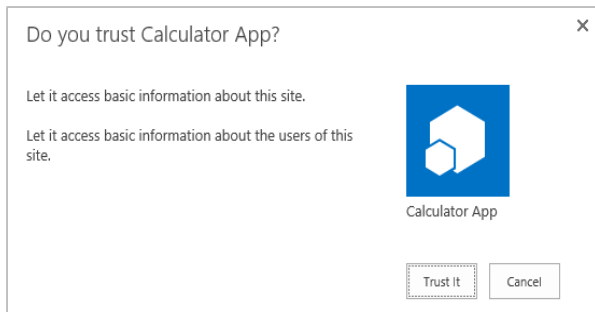
Installing Apps

- App installer must be site administrator
 - Click **add an app** link on **Site Contents** page
 - **add an app** link takes you to app discovery page (*addanapp.aspx*)
 - On **Your apps** page, click on app tile to install that app

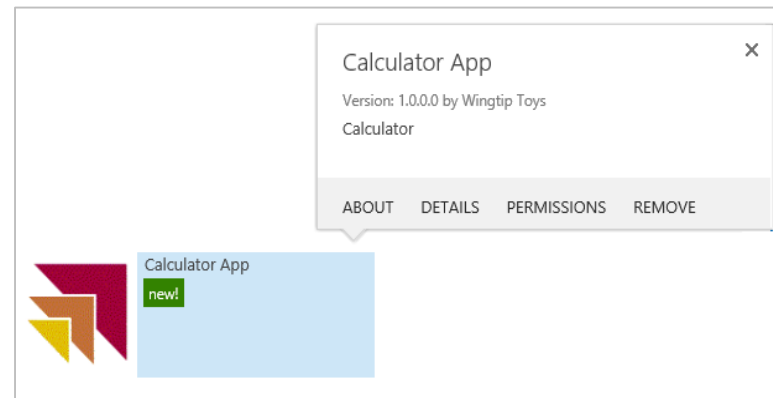
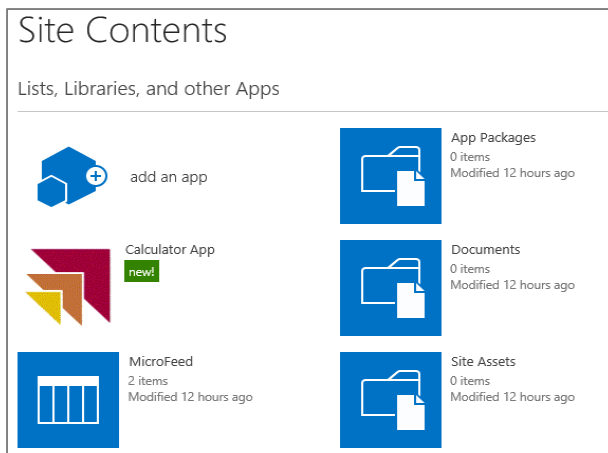


Completing the App installation Process

- App installer is Prompted to Explicitly Trust the app During Installation

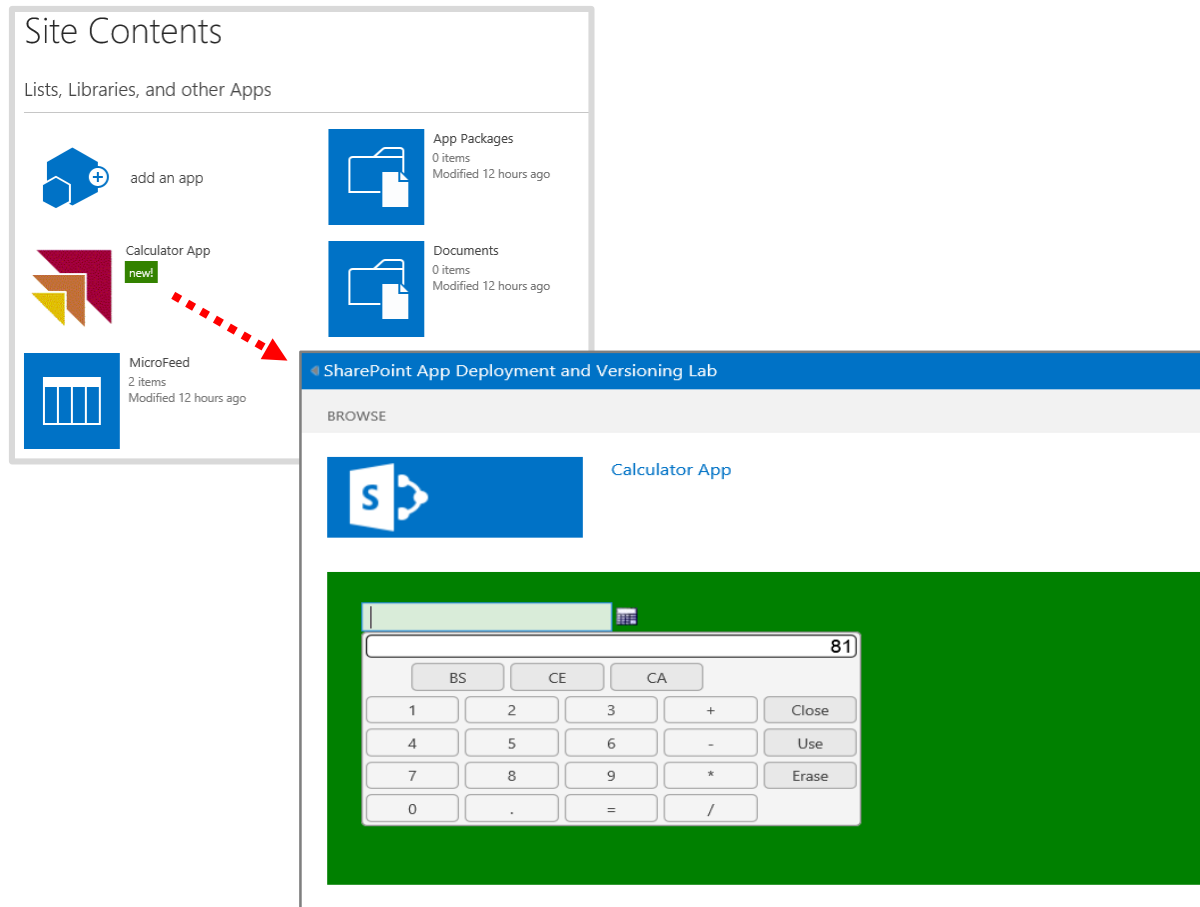


- After installation, app tile added to **Site Contents** page
app tile provides fly-out menu to assist with app management



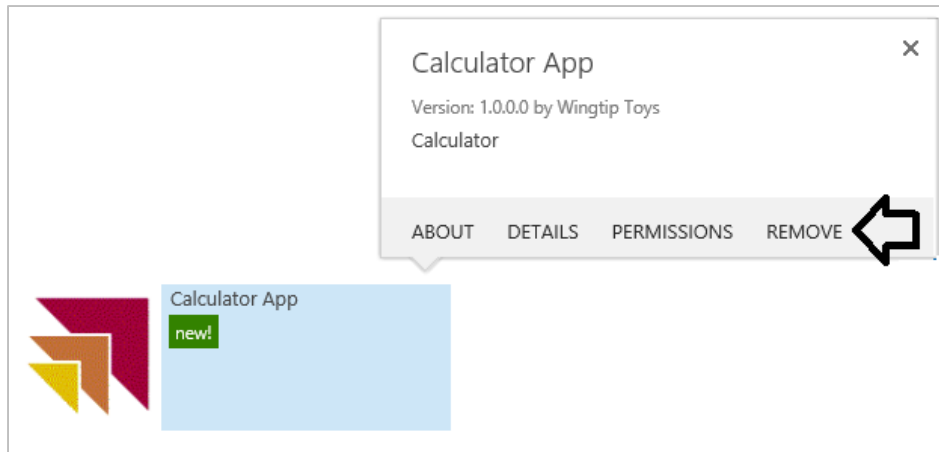
Launching the App After Installation

- Clicking app tile on Site Contents page launches app
 - User is redirected to app start page



Uninstalling an App

- Use Remove command in app tile flyout menu
 - SharePoint deletes App Web if there is one



The background of the slide is a close-up, low-angle shot of a server rack. The rack is filled with numerous server units, each featuring a grid of small, glowing blue lights. The perspective is looking up the length of the rack, creating a sense of depth and scale. The lighting is predominantly blue, giving it a high-tech, digital feel.

DEMO

Publishing and Installing a SharePoint-hosted App

Installing Apps at Tenancy Scope

- App catalog site used for tenancy-scoped install
 - Step 1: Publish app to app catalog site
 - Step 2: Install app in same app catalog site
 - Step 3: Configure app to make it available in other sites

The screenshot shows the 'Calculator App' deployment configuration interface. At the top, a modal window displays the app details: 'Calculator App', 'Version: 1.0.0.0 by Wingtip Toys', and 'Calculator'. Below this, there are two tabs: 'ABOUT' and 'DEPLOYMENT'. A yellow callout box with a black arrow points to the 'DEPLOYMENT' tab, stating: 'Deployment command only appears for apps installed in app catalog sites'. Below the modal, a red dotted arrow points from the 'DEPLOYMENT' tab to the 'Manage App Deployments' section. This section is titled 'Manage App Deployments · Calculator App' and includes a sub-header: 'Select the site collections, managed paths, or site templates where you want this app to appear. It may take up to 30 minutes for users to see these changes on their sites.' Under the 'Site Collections' heading, there is a text input field labeled 'Enter a site collections to deploy to:' and an 'Add' button. Below this, there is another text input field labeled 'Site collections to deploy to:' and a 'Remove' button.

Configuring an App Deployment

- App Deployment can be configured in terms of managed paths...

The 'Managed Paths' configuration dialog is shown. It has three main sections: 'Managed Paths' on the left, 'Available managed paths:' in the center, and 'Managed paths to deploy to:' on the right. The 'Managed Paths' section contains the text: 'Use this section to specify what managed paths should have this app available.' The 'Available managed paths:' section contains a list box with the item 'sites'. Below this list box is the label 'Managed path:'. Between the two list boxes are two buttons: 'Add >' and '< Remove'. The 'Managed paths to deploy to:' section contains a list box with the item '(All Paths)' selected.

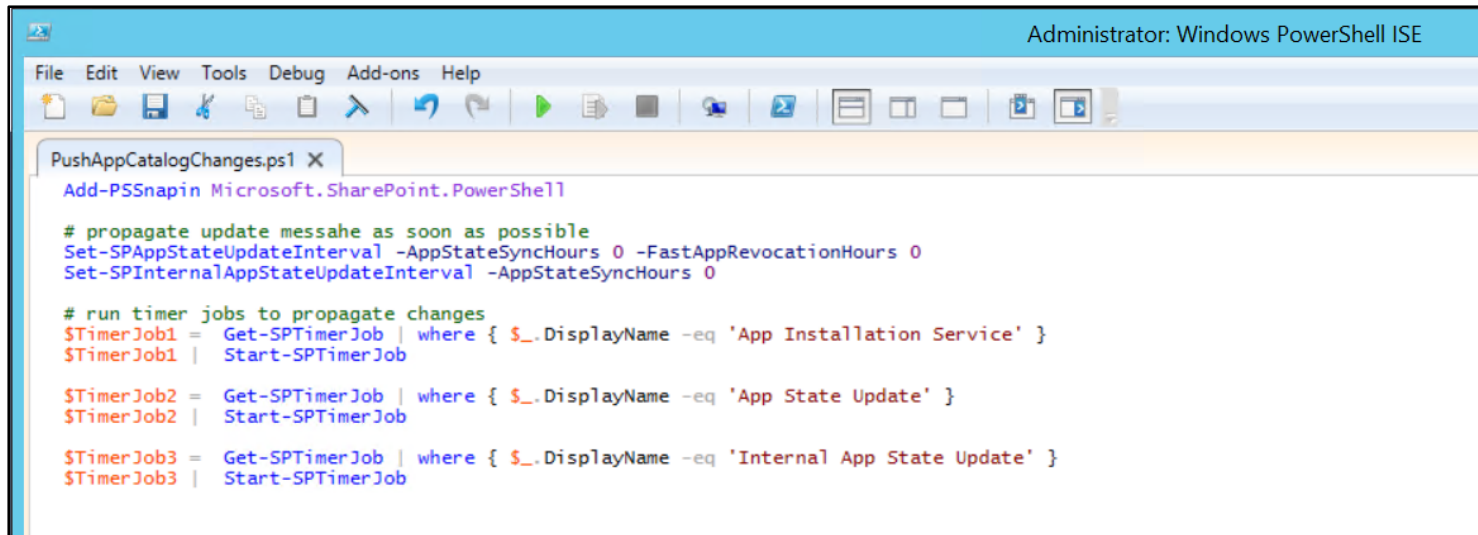
- App Deployment can be configured in terms of Site templates...

The 'Site Templates' configuration dialog is shown. It has three main sections: 'Site Templates' on the left, 'Available site templates:' in the center, and 'Site templates to deploy to:' on the right. The 'Site Templates' section contains the text: 'Use this section to specify what site templates should have this app available.' The 'Available site templates:' section contains a list box with the following items: 'Academic Library', 'Basic Search Center', 'Blog', 'Business Intelligence Center', 'Community Portal', 'Community Site', 'Document Center', 'eDiscovery Case', 'eDiscovery Center', and 'Enterprise Search Center'. Below this list box is the label 'Description: A site for developers to build, test and publish apps for Office'. Between the two list boxes are two buttons: 'Add >' and '< Remove'. The 'Site templates to deploy to:' section contains a list box with the items 'Team Site' and 'Developer Site', with 'Developer Site' selected. At the bottom right of the dialog are two buttons: 'OK' and 'Cancel'.



Pushing out App Deployment Changes

- Deployment changes are not instant
 - Usually requires time job to push changes to all sites
 - Updates can take up to 24 hours to propagate
 - PowerShell script can be used to speed things up for testing



The screenshot shows the Windows PowerShell ISE interface. The title bar reads "Administrator: Windows PowerShell ISE". The menu bar includes File, Edit, View, Tools, Debug, Add-ons, and Help. The toolbar contains various icons for file operations and execution. The script editor shows a file named "PushAppCatalogChanges.ps1". The script content is as follows:

```
Add-PSSnapin Microsoft.SharePoint.PowerShell

# propagate update message as soon as possible
Set-SPAppStateUpdateInterval -AppStateSyncHours 0 -FastAppRevocationHours 0
Set-SPInternalAppStateUpdateInterval -AppStateSyncHours 0

# run timer jobs to propagate changes
$TimerJob1 = Get-SPTimerJob | where { $_.DisplayName -eq 'App Installation Service' }
$TimerJob1 | Start-SPTimerJob

$TimerJob2 = Get-SPTimerJob | where { $_.DisplayName -eq 'App State Update' }
$TimerJob2 | Start-SPTimerJob

$TimerJob3 = Get-SPTimerJob | where { $_.DisplayName -eq 'Internal App State Update' }
$TimerJob3 | Start-SPTimerJob
```



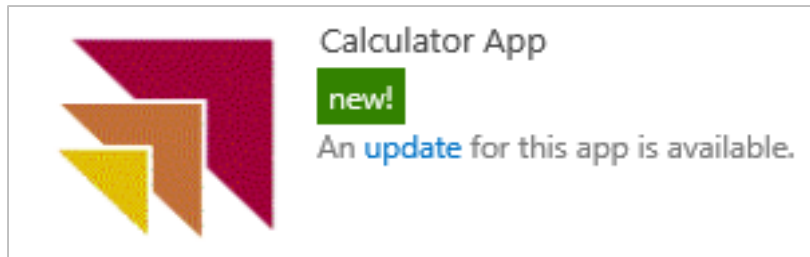
The background of the slide is a close-up, low-angle shot of a server rack. The rack is filled with numerous server units, each featuring a grid of small, glowing blue lights. The perspective is looking up the length of the rack, creating a sense of depth and scale. The lighting is predominantly blue, giving it a high-tech, digital feel.

DEMO

Installing an App at Tenancy Scope

Upgrading Apps

- Overview of SharePoint app upgrade process
 - App catalog tracks current version number of app
 - SharePoint host records app version at install time
 - Updated version of app can be uploaded to app catalog
 - SharePoint host notifies user when there's new version

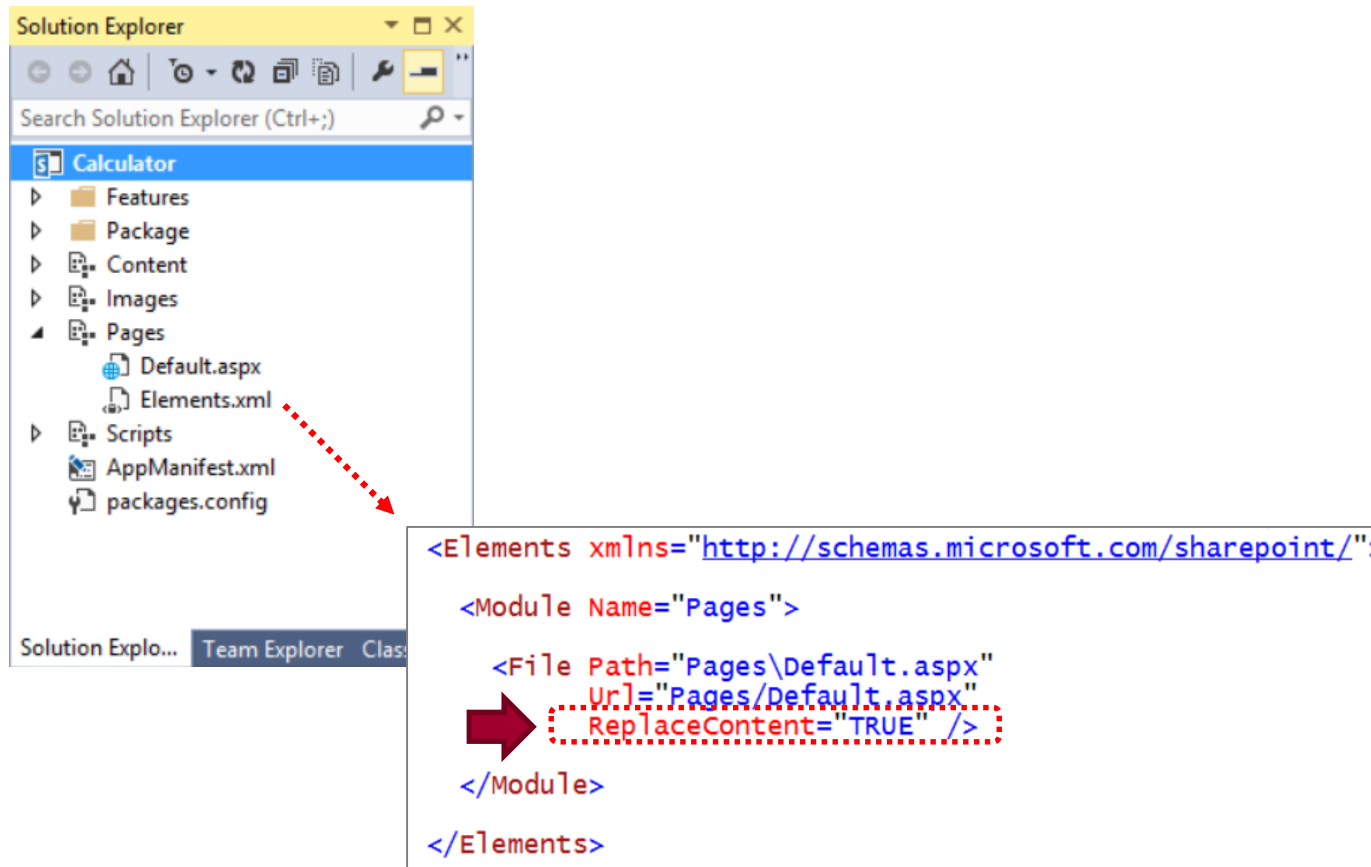


- User makes choice whether upgrade or not
 - User (app installer) is never forced to upgrade
 - If user decides to upgrade, upgrade involves clicking button



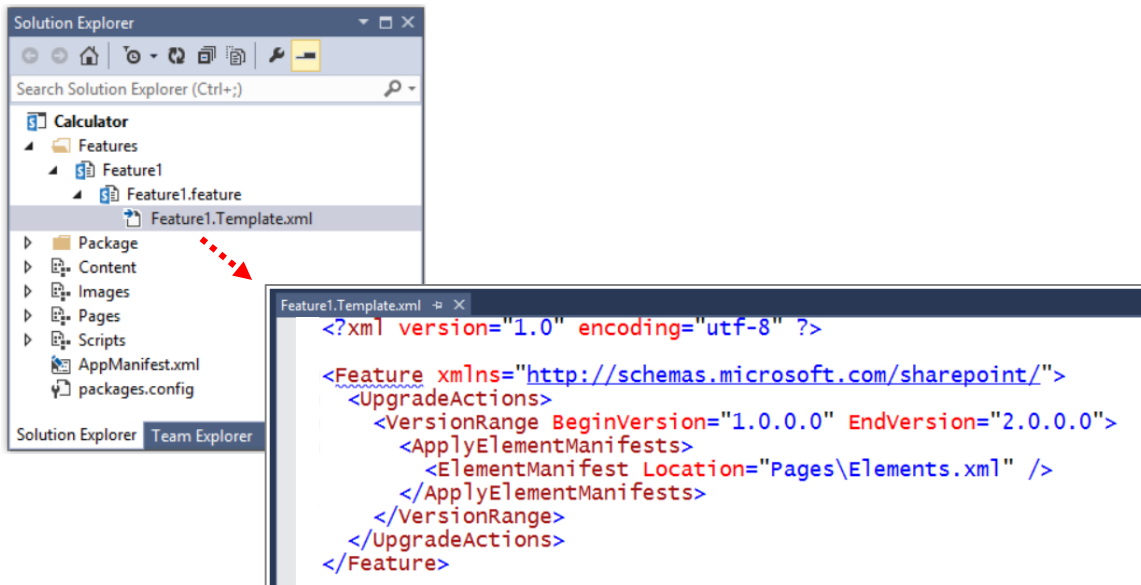
Overwriting Existing Files During Upgrade

- Files to be upgrade required **ReplaceContent** attribute
 - Edit must be made by hand to **Elements.xml** file

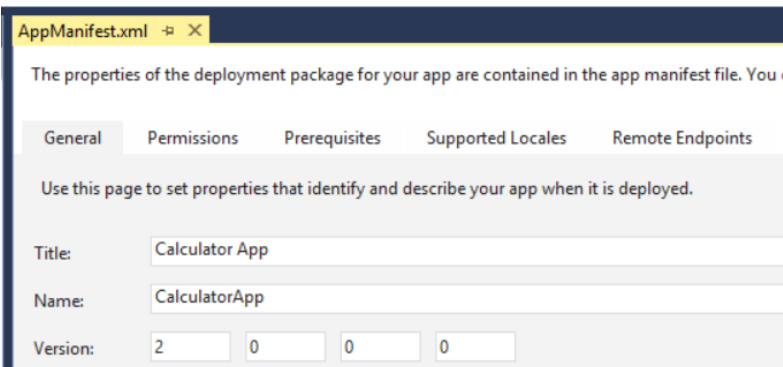


Configuring Feature Upgrade Actions

- Upgrade actions must be entered by hand



- Make sure to update version number in AppManifest.xml





DEMO

Upgrading the Start Page of a SharePoint-hosted App

Agenda

- ✓ Creating the App Catalog
- ✓ Publishing Apps in the App Catalog
- ✓ Installing and Upgrading Apps
- Deploying Provider-hosted Apps



Creating an SSL Certificate

- Test SSL certificate can be created & installed using PowerShell

```
CreateCertificateForWingtipSearchApp.ps1 X
$makecert = "C:\Program Files\Microsoft Office Servers\15.0\Tools\makecert.exe"
$certmgr = "C:\Program Files\Microsoft Office Servers\15.0\Tools\certmgr.exe"

# specify subject for SSL certificate
$subject = "searchapp.wingtip.com"

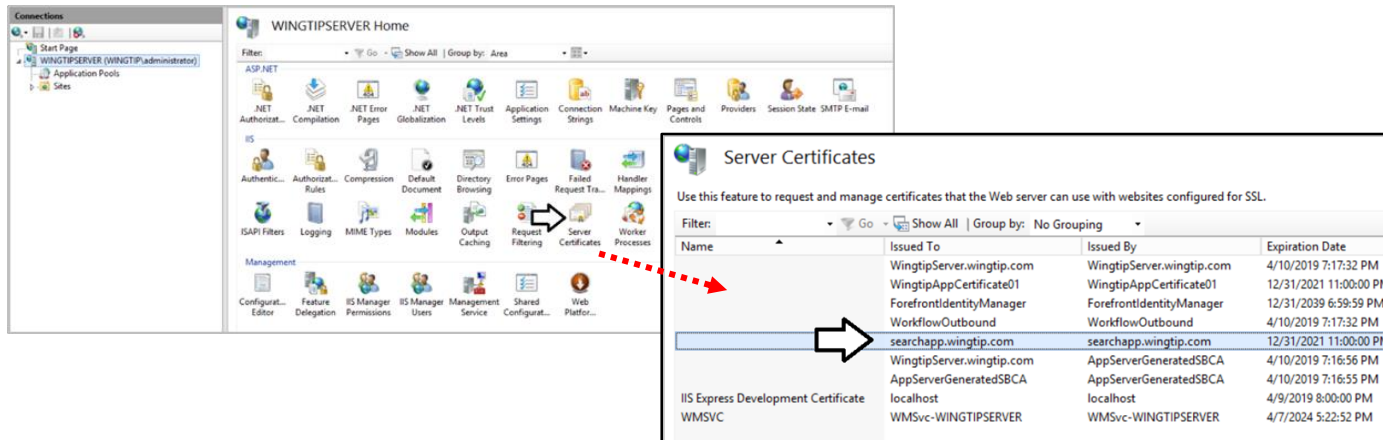
# create output directory to create SSL certificate file
$outputDirectory = "c:\Certs\"
New-Item $outputDirectory -ItemType Directory -Force -Confirm:$false | Out-Null

# create file name for SSL certificate files
$publicCertificatePath = $outputDirectory + $subject + ".cer"

Write-Host
Write-Host "Creating .cer certificate file..."
& $makecert -r -pe -n "CN=$subject" -b 01/01/2012 -e 01/01/2022 -eku 1.3.6.1.5.5.7.3.1 -ss my -sr localMachine root

Write-Host
Write-Host "Registering certificate with IIS..."
& $certmgr /add $publicCertificatePath /s /r localMachine root
```

- Use IIS Administrative tool to further configure SSL certificate



Server Certificates

Use this feature to request and manage certificates that the Web server can use with websites configured for SSL.

Name	Issued To	Issued By	Expiration Date
WingtipServer.wingtip.com	WingtipServer.wingtip.com	WingtipServer.wingtip.com	4/10/2019 7:17:32 PM
WingtipAppCertificate01	WingtipAppCertificate01	WingtipAppCertificate01	12/31/2021 11:00:00 PM
ForefrontIdentityManager	ForefrontIdentityManager	ForefrontIdentityManager	12/31/2039 6:59:59 PM
WorkflowOutbound	WorkflowOutbound	WorkflowOutbound	4/10/2019 7:17:32 PM
searchapp.wingtip.com	searchapp.wingtip.com	searchapp.wingtip.com	12/31/2021 11:00:00 PM
WingtipServer.wingtip.com	WingtipServer.wingtip.com	AppServerGeneratedSBCA	4/10/2019 7:16:56 PM
AppServerGeneratedSBCA	AppServerGeneratedSBCA	AppServerGeneratedSBCA	4/10/2019 7:16:55 PM
localhost	localhost	localhost	4/9/2019 8:00:00 PM
IIS Express Development Certificate	localhost	localhost	4/7/2024 5:22:52 PM
WMSVC	WMSvc-WINGTIPSERVER	WMSvc-WINGTIPSERVER	4/7/2024 5:22:52 PM

Creating the IIS Website

- Creating an IIS Website
 - Can be done manually
 - Can be done using a PowerShell script

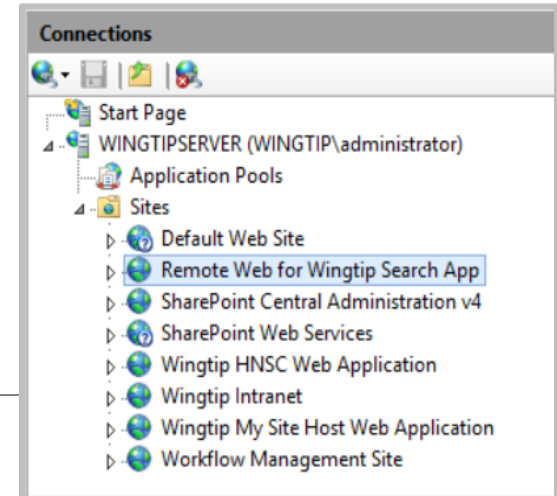
```
$IisWebSiteName = "Remote Web for Wingtip Search App"
$IisWebSitePath = "C:\inetpub\WingtipSearchApp"
$hostHeader = "searchapp.wingtip.com"
$bindingPort = 80
$appPoolName = "DefaultAppPool"
$IisWebSite = Get-Website | Where-Object {$_.Name -eq $IisWebSiteName}

if ($IisWebSite -ne $null) { Write-Host "    IIS Web site named $IisWebSiteName has already created" -ForegroundColor Gray }
else {
    #check for existing physical path
    if (Test-Path $IisWebSitePath) {}
    else{
        $silence = New-Item -Path $IisWebSitePath -ItemType Directory
        Write-Host "    Path $IisWebSitePath created" -ForegroundColor Gray
    }
}

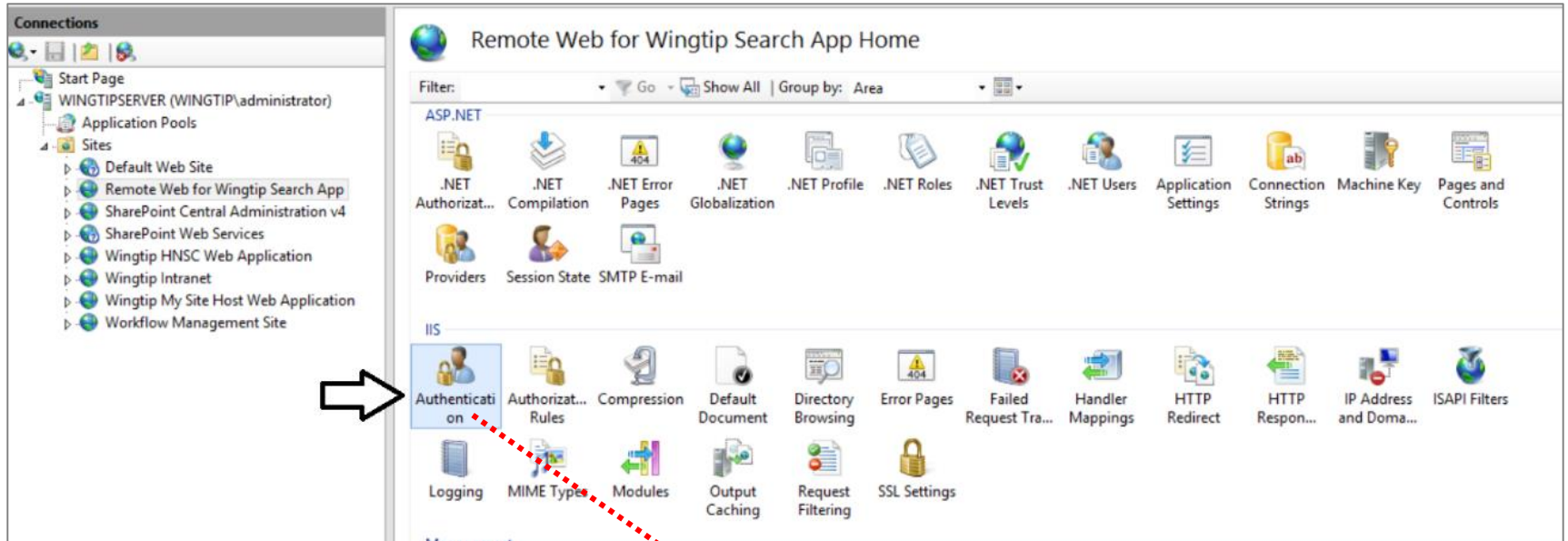
Write-Host "    .. creating IIS web site: $IisWebSiteName..." -ForegroundColor Gray
$IisWebSite = New-Website -Name $IisWebSiteName -PhysicalPath $IisWebSitePath -ApplicationPool $appPoolName -HostHeader $hostHeader -Port $bindingPort
Write-Host "    $IisWebSiteName site created" -ForegroundColor Gray
}

# add entry to HOST file to fix Visual Studio bug
$hostsFilePath = "c:\Windows\System32\Drivers\etc\hosts"
$hostFileEntry = "127.0.0.1    $hostHeader"
Add-Content -Path $hostsFilePath -Value "`r`n$hostFileEntry"
Write-Host "HOST file entry added: $hostFileEntry" -ForegroundColor Gray

Write-Host
Write-Host "Finished! You can now import web applications." -ForegroundColor Green
```



Configuring Windows Authentication

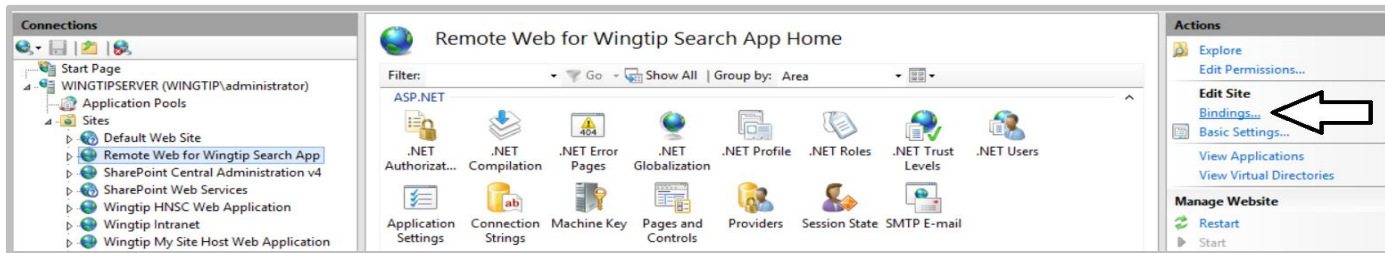


Name	Status	Response Type
Anonymous Authentication	Enabled	
ASP.NET Impersonation	Disabled	
Basic Authentication	Disabled	HTTP 401 Challenge
Digest Authentication	Disabled	HTTP 401 Challenge
Forms Authentication	Disabled	HTTP 302 Login/Redirect
Windows Authentication	Enabled	HTTP 401 Challenge

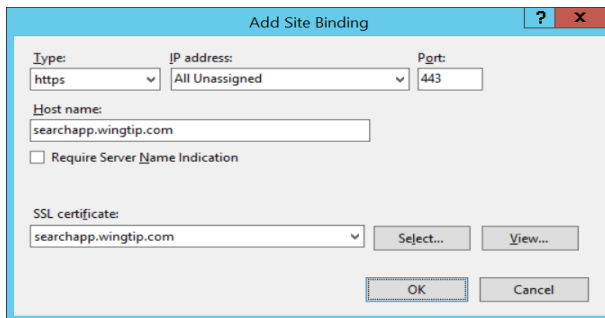


Configuring SSL for the Remote Web

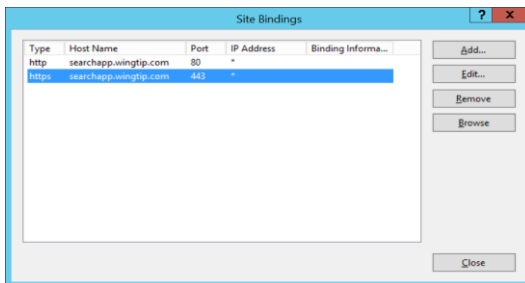
- Editing bindings to the associated IIS Website



- Create a new HTTPS binding and associate with SSL certificate



- Binding listens for requests targeting **http://searchapp.wingtip.com**





DEMO

Creating and Configuring an IIS Website to Host the Remote Web

Registering the App Security Principal

- Can be done manually using AppRegNew.aspx page
 - Enter the **App ID** (aka Client ID)
 - Enter an **App Secret** (even in S2S scenario when it is not used)
 - Enter a **Title**
 - Enter an **App Domain**

App Information
The app's information, including app id, secret, title, hosting url and redirect url.

App Id:

App Secret:

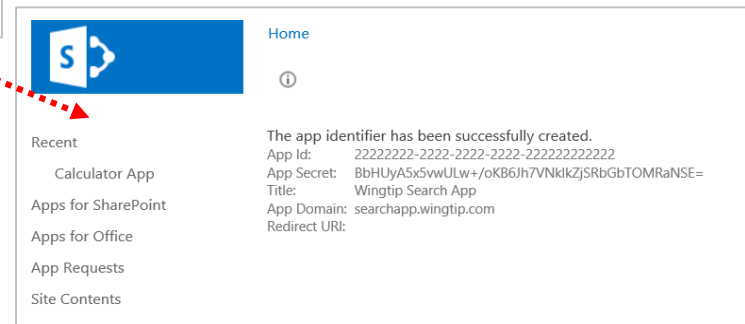
Title:

App Domain:

Example: "www.contoso.com"

Redirect URL:

Example: "https://www.contoso.com/default.aspx"



The screenshot shows the SharePoint 'Home' page. On the left, under the 'Recent' section, there is a list of items: 'Calculator App', 'Apps for SharePoint', 'Apps for Office', 'App Requests', and 'Site Contents'. On the right, a confirmation message states: 'The app identifier has been successfully created.' Below this message, the following details are listed: 'App Id: 22222222-2222-2222-2222-222222222222', 'App Secret: BbHUyA5x5vwULw+/oKB6Jh7VNklkZJSRbGbTOMRaNSE=', 'Title: Wingtip Search App', 'App Domain: searchapp.wingtip.com', and 'Redirect URI:'. A red dotted arrow points from the 'Create' button in the 'App Information' form to the 'Recent' list on the 'Home' page.

Registering App Principal with PowerShell

- Two ways to do this
 - The easy way using **Register-SPAppPrincipal**
 - The more flexible way using **SPAppPrincipalManager**

```
Add-PSSnapin "Microsoft.SharePoint.PowerShell"

# set initialization values for new app principal
$appDisplayName = "My First S2S App"
$clientID = "22222222-2222-2222-2222-222222222222"

$targetSiteUrl = "http://wingtipserver"
$targetSite = Get-SPSite $targetSiteUrl
$realm = Get-SPAAuthenticationRealm -ServiceContext $targetSite
$fullAppPrincipalIdentifier = $clientID + '@' + $realm

Write-Host "Register new app principal"
$registeredAppPrincipal = Register-SPAppPrincipal -NameIdentifier $fullAppPrincipalIdentifier
                                           -Site $targetSite.RootWeb
                                           -DisplayName $appDisplayName

$registeredAppPrincipal | select * | Format-List
$registeredAppPrincipal | select * | Format-List | Out-File -FilePath "SecurityPrincipal01.txt"
```



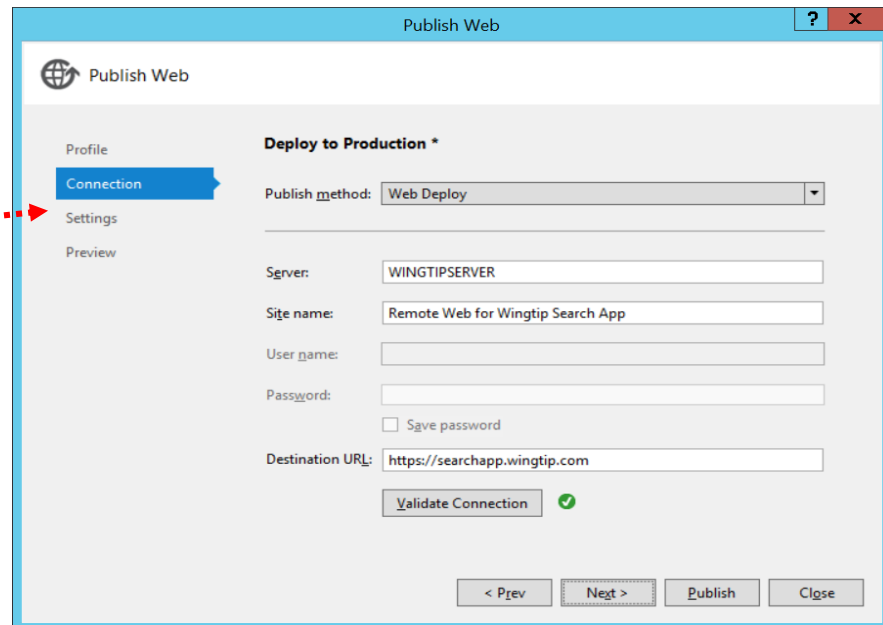
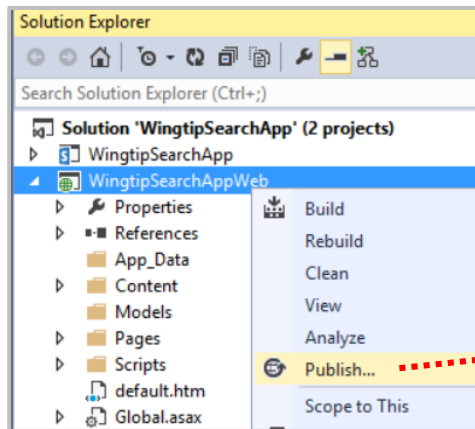


DEMO

Registering the App Security Principal for a Provider-hosted App

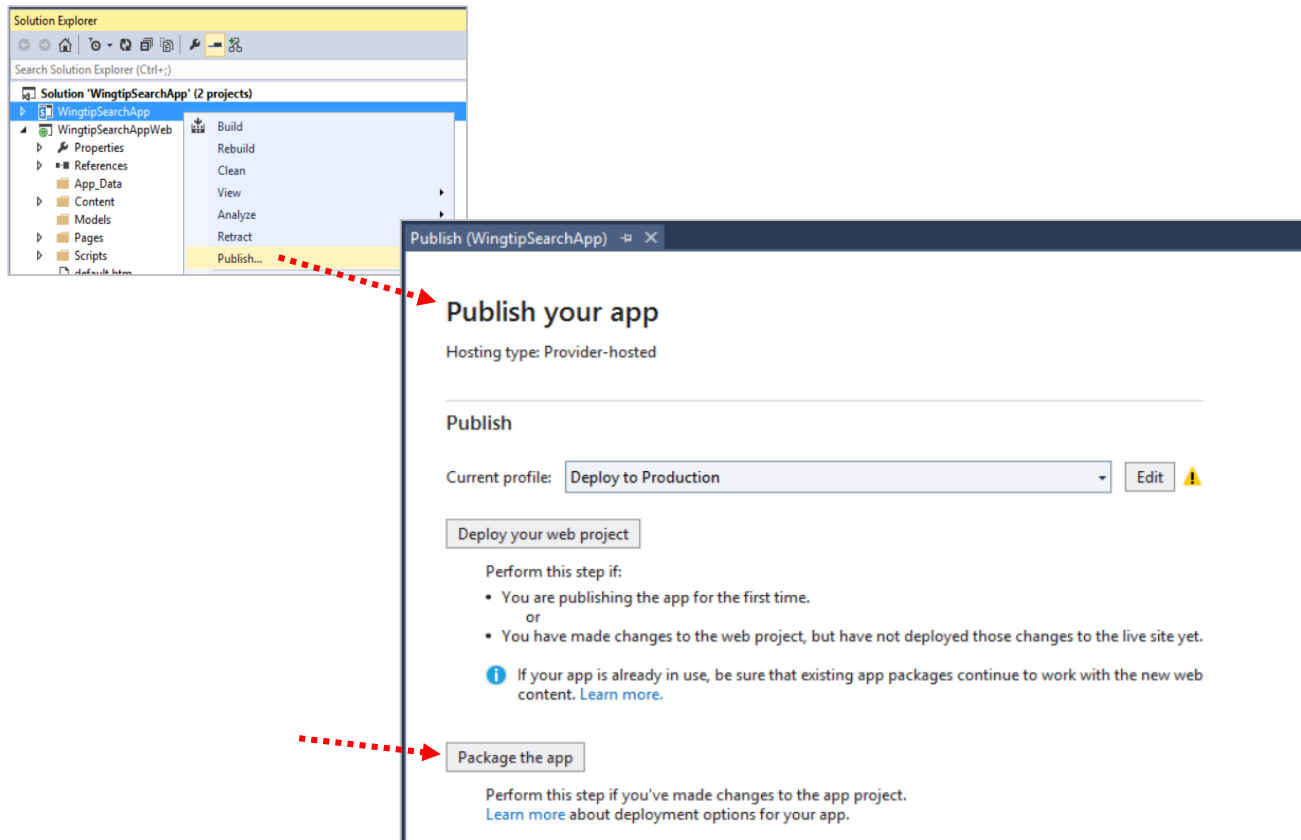
Deploying the App's Remote Web Project

- Visual Studio provides Publish Web wizard
 - Allows you to create one or more deployment configurations
 - Automatically pushes Web Project files into IIS website



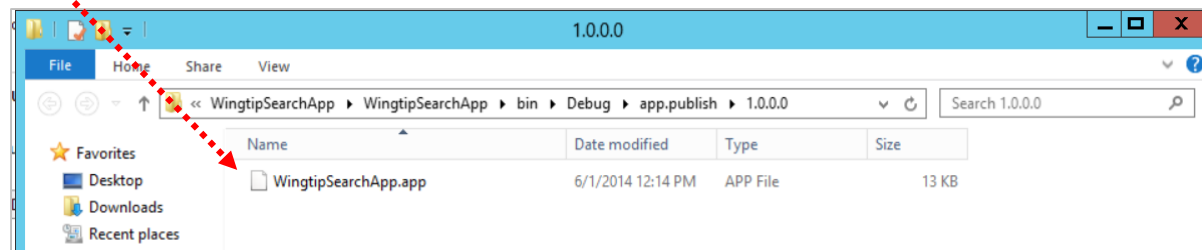
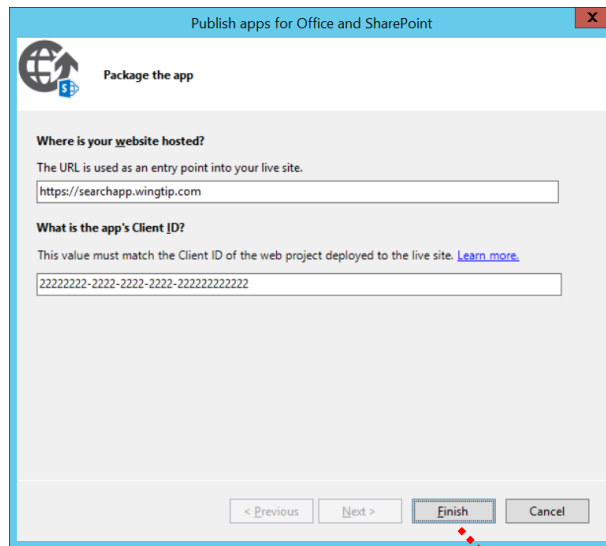
Creating the Provider-hosted App Package

- Visual Studio provides Publish your app page
 - Click **Package the app** to begin packaging process



Enter Information for the App Package

- Provide-hosted App must be published with...
 - SSL-based URL where the remote web has been deployed
 - The App ID (aka Client ID)



Publishing and Installing the App

- Publish the app package *(just like with a SharePoint-hosted app)*

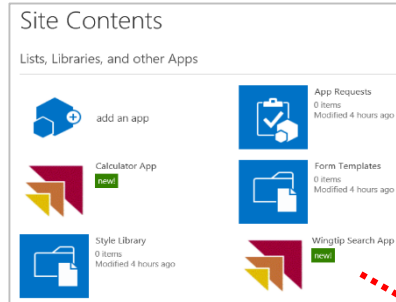
Apps for SharePoint ⓘ

⊕ new app or drag files here

All Apps Featured Apps Unavailable Apps ...

✓	□	Title	Name	App Version	Edit	Product ID	Metadata Language	Default Metadata Language
Product ID : {921FDA40-576C-48CF-B00B-7285D24372A0} (1)								
□		Wingtip Search App	WingtipSearchApp	1.0.0.0		{921FDA40-576C-48CF-B00B-7285D24372A0}	English - 1033	Yes
Product ID : {2AB41D1E-C8A8-400A-AD6E-7C15FEE6A69A} (1)								
□		Calculator App	Calculator	2.0.0.0		{2AB41D1E-C8A8-400A-AD6E-7C15FEE6A69A}	English - 1033	Yes

- Install and launch



Host Web

Wingtip Search App

Find Top-level Sites Find Lists Find Tasks General Search ContentClass:STS_SITE

Search Result Title	Search Result URL
Wingtip Intranet	http://intranet.wingtip.com
SharePoint App Deployment and Versioning Lab	http://publishing.wingtip.com
Wingtip Toys	http://www.wingtip.com
Wingtip App Catalog	http://wingtipserver/sites/AppCatalog
Wingtip BI Center	http://intranet.wingtip.com/sites/bicenter
SP_Crawler	http://my.wingtip.com/personal/sp_crawler



DEMO

Deploying a Provider-hosted App in an On-premises Farm

Summary

- ✓ Creating the App Catalog
- ✓ Publishing Apps in the App Catalog
- ✓ Installing and Upgrading Apps
- ✓ Deploying Provider-hosted Apps

