



SharePoint 2013 and SharePoint Online solution pack for branding and site provisioning

Microsoft Corporation

July 2014

**Applies to:** SharePoint 2013 and SharePoint Online

**Summary:** This solution pack includes code and documents that demonstrate and describe site branding and site provisioning techniques for SharePoint 2013 and SharePoint Online.

©2014 Microsoft Corporation. All rights reserved.

This document is provided "as-is." Information and views expressed in this document, including URL and other Internet website references, may change without notice. You bear the risk of using it.

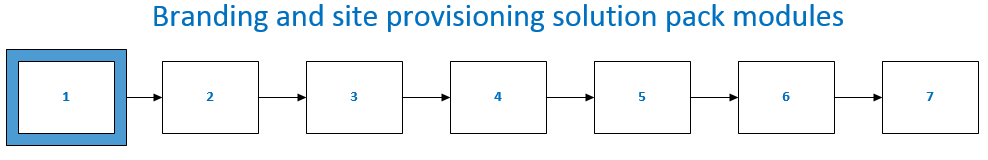
Some examples are for illustration only and are fictitious. No real association is intended or inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes. You may modify this document for your internal, reference purposes.

Module 1: SharePoint 2013 site branding and provisioning 101

# What is a solution pack?

A solution pack is a set of evolving guidance created in a partnership between the SharePoint product team and the SharePoint community. Solution packs are aligned around a specific scenario and exist to clarify complicated stories and help users meet goals with new patterns and technologies.



The core building block of each solution pack is the *module*. The scope and purpose of the module remains constant, but the content within each module can grow and change over time. This is version 1.0 of a solution pack, which contains five module documents, associated code samples, inline code examples, and numerous links to additional Microsoft and SharePoint community resources.

Some examples of content types that can be included in a solution pack:

* **Documents**. Written guidance you can use to understand concepts, choose from among options, and complete tasks.
* **Pictures**. Visual elements such as diagrams and infographics that present complex information simply and clearly.
* **Code**. Code samples that do complicated customization work for you or provide an easily customizable resource you can use to save coding and design time.
* **Case studies**. Case studies that describe how others in the same situation used new technologies and patterns to achieve business and design goals.

# How is this different?

Historically, Microsoft has published product documentation that has come mostly from the product team, and we’ve provided guidance to enterprise customers largely from the field via consulting services.

With solution packs, the product team is engaging directly with consulting services and the community to use the cycle of services consumption and production to deliver a dynamic story that cycles around customer needs and goals.

**Why a solution pack for SharePoint 2013 site branding and provisioning?**

With the introduction of the cloud app model (CAM) and apps for SharePoint, there are now viable alternatives to older, more established ways of branding and provisioning SharePoint sites.

This solution pack supports all phases of site branding and emphasizes using apps for SharePoint to provision site branding throughout the enterprise and manage branding at scale: a pattern known as [*remote provisioning*](http://blogs.msdn.com/b/vesku/archive/2013/08/23/site-provisioning-techniques-and-remote-provisioning-in-sharepoint-2013.aspx) (Vesa Juvonen). While SharePoint developers have historically used the SharePoint feature framework, site templates, web templates, and site definitions to provision sites and site collections, the remote provisioning pattern shows you how to create custom [apps for SharePoint](http://msdn.microsoft.com/en-US/office/dn448479) that provision site branding and perform other site provisioning tasks.

Some sections of the solution pack documents provide capsule overviews and introductions to content published elsewhere, such as on TechNet, MSDN, and SharePoint community resources. Where documentation does not currently exist on Microsoft sites, the solution pack provides link to additional resources that help fill gaps, and provides new content and code to show how you can use CAM and the apps for SharePoint model.

**Acknowledgements**

The solution pack was made possible by contributions and/or review by the following individuals:

*Cathy Anderson, Yina Arenas, Todd Baginski, Mike Chorey, Jim Crowley, Lincoln Demaris, Richard di Zerega, Kimmo Forss, Ken Harder, Jason Hershey, Rob Howard, Randall Isenhour, Bert Jansen, Vesa Juvonen, Jonathan Keslin, Frank Marasco, AJ May, Tim McDaniel, Ken Milne, Anna Ngo, Matjaz Perpar, Lionel Robinson, Mike Russell, Michael Sherman, Thiru Srinivasan, Kiki Shuxteau, Rohit Tripathi, Koen Vosters, and Steve Walker*

# Contents

The SharePoint 2013 and SharePoint Online solution pack includes five documents, or modules, listed in Table 1.

**Table 1. SharePoint site provisioning and branding solution pack modules**

|  |  |  |  |
| --- | --- | --- | --- |
| **Module** | **Name** | **Description** | |
| 1 | SharePoint site branding and remote provisioning 101 | Introduces the solution pack concept and contents, including SharePoint site fundamentals and the branding and site provisioning workflow. It also introduces the remote provisioning pattern, a new way to use apps for SharePoint and the Cloud App Model (CAM) to provision branding to SharePoint sites. | |
| 2 | Inside SharePoint Pages | Presents the SharePoint page model and its component parts. | |
| 3 | Building SharePoint sites and pages | Describes tooling options for developing site branding elements and managing them in SharePoint. | |
| 4 | Customizing site branding | Covers composed looks, the SharePoint 2013 theming engine, and best practices for customizing the look and feel of your site with custom CSS and CSOM via remote provisioning. | |
| 5 | Using apps for SharePoint to provision SharePoint site branding | Describes the remote provisioning pattern, and shows how to apply site branding at the site collection, site, and subsite levels by using the pattern. Demonstrates how to set up remote provisioning to use a custom form, an approval workflow, and refresh tokens. | |
| 6 | Migrating full trust code (FTC) solutions to the app model | Provides guidance, tips, and code for analyzing current FTC solutions and, when applicable, specific guidance for migrating specific solutions to the app model. | |
| 7 | Site navigation and branding publishing sites | Describes CSOM capabilities for site navigation, and explores branding considerations specific to Publishing sites. | |
|  | | |  |

## Where are the samples?

**Important** Each module document is associated with one or more code samples, which are downloadable individually from Code Gallery or together in the SharePoint 2013 and SharePoint Online branding and site provisioning solution pack’s [sample pack](http://code.msdn.microsoft.com/SharePoint-2013-Sample-81b03d1e).

# Key Terms and Concepts

Table 2 lists terms and concepts that are useful to know as you start to work with SharePoint site provisioning and branding with the remote provisioning pattern.

**Table 2. Key terms and concepts**

|  |  |  |  |
| --- | --- | --- | --- |
| **Term** | **Definition** | **File system, path, syntax, or UI location** | **Guidance** |
| [App for SharePoint](http://msdn.microsoft.com/library/office/fp179930.aspx) |  | Site Contents tab | [Build apps for SharePoint](http://msdn.microsoft.com/en-us/library/office/apps/jj163230.aspx)  [Apps for SharePoint sample pack](http://code.msdn.microsoft.com/sharepoint/Apps-for-SharePoint-sample-64c80184) |
| App web | The website from which an app is deployed. | Structure of an app web URL:  *https:// App\_Prefix - App\_ID . App\_Base\_Domain / Domain\_Relative\_URL\_of\_Host\_Web / App\_Name* | [Host webs, app webs, and SharePoint components](http://msdn.microsoft.com/en-us/library/office/fp179925.aspx) |
| Approval workflow | Workflows specific to Publishing sites that specifies who approves the publication of a page pending publication, and when. |  | [SharePoint 2013 approval workflow](http://blogs.msdn.com/b/thirusrinivasan1/archive/2013/10/31/sharepoint-2013-workflow-calling-a-wcf-service.aspx) (Thiru Srinivasan)  [Get started with workflows in SharePoint 2013](http://msdn.microsoft.com/en-us/library/office/jj163917.aspx) |
| ClientContext | A central object that serves as a “center of gravity” for all SharePoint CSOM and JSOM operations. | public class ClientContext : ClientRuntimeContext | [ClientContext](http://msdn.microsoft.com/en-us/library/office/microsoft.sharepoint.client.clientcontext.aspx) class |
| Cloud App Model (CAM) | Apps for SharePoint are self-contained pieces of functionality that extend the capabilities of a SharePoint website. You can use CAM to author and deliver secure, performant, flexible, and consistent apps for SharePoint. |  | [Apps for SharePoint overview](http://msdn.microsoft.com/en-us/library/office/fp179930.aspx) |
| Content database | Content databases store all content for a site collection. | The content database stores the following kind of data:   * Site documents and files in document libraries * List data * Web Part properties * Sandboxed solutions * User names and rights | Content databases are attached to site collections (one per site collection automatically; others can be attached), and to a SharePoint web application.  More about the SharePoint 2013 content database: [Database types and descriptions](http://technet.microsoft.com/en-us/library/cc678868.aspx) |
| CSOM | Client-side object model. A model for writing client-side code for SharePoint with the .NET framework. |  | [Using the client-side object model (CSOM)](http://msdn.microsoft.com/en-us/library/ff798388.aspx)  [[MS-CSOM]: SharePoint Client Query Protocol](http://msdn.microsoft.com/en-us/library/dd912094(v=office.15).aspx)  [SharePoint 2013 .NET Server, CSOM, JSOM, and REST API index](http://msdn.microsoft.com/en-us/library/office/dn268594.aspx) |
| File system |  |  | See also “Hive” and “content database.” |
| Hive | SharePoint’s physical files; the files in the file system. These files are distinct from content stored in a content database. | %program files%/Common Files/Microsoft Shared/Web Server Extensions/15/ | See also “File system” and “content database.” |
| Host web | The website on which an app is installed. |  | [Host webs, app webs, and SharePoint components](http://msdn.microsoft.com/en-us/library/office/fp179925.aspx) |
| MySite | See OneDrive for Business. |  |  |
| [OneDrive for Business](http://office.microsoft.com/en-us/sharepoint-server-help/what-is-skydrive-pro-HA102822076.aspx) | A personal library for storing and organizing work documents that are shareable within your organization. |  | [OneDrive for Business is different from OneDrive](http://office.microsoft.com/en-us/sharepoint-server-help/what-is-skydrive-pro-HA102822076.aspx#differences)  [OneDrive for Business is different from your team site](http://office.microsoft.com/en-us/sharepoint-server-help/redir/HA104105232.aspx?CTT=5&origin=HA102822076)  [On-demand provisioning of personal sites on Office 365](http://blogs.msdn.com/b/vesku/archive/2013/11/25/office365-apply-automatically-custom-branding-to-personal-site-skydrive-pro.aspx) (Vesa Juvonen) |
| Personal sites | See OneDrive for Business. |  |  |
| Remote provisioning | A model that provisions sites by using templates and code that runs outside SharePoint in a provider-hosted app. |  | [Site provisioning techniques and remote provisioning in SharePoint 2013](http://blogs.msdn.com/b/vesku/archive/2013/08/23/site-provisioning-techniques-and-remote-provisioning-in-sharepoint-2013.aspx) (Vesa Juvonen)  [Self-service site provisioning using apps for SharePoint 2013](http://blogs.msdn.com/b/richard_dizeregas_blog/archive/2013/04/04/self-service-site-provisioning-using-apps-for-sharepoint-2013.aspx) (Richard di Zerega) |
| REST | REpresentional State Transfer. A stateless architectural style that abstracts architectural elements and uses HTTP verbs read and write data from Web pages that contain XML files. |  | [Get Started with the SharePoint 2013 REST service](http://msdn.microsoft.com/en-us/library/office/apps/fp142380.aspx)  [Using the REST interface](http://msdn.microsoft.com/en-us/library/ff798339.aspx) |
| Root web | The first web inside of a site collection. |  | The root web is also sometimes referred to as the “Web Application Root.” |
| [SharePoint Online](http://office.microsoft.com/en-us/sharepoint/sharepoint-online-online-collaboration-software-FX103789366.aspx) | SharePoint Online is the cloud-based SharePoint offering in Office 365. |  |  |
| Site | A group of sites that share the same owner and administrative settings, such as permissions. |  | [Web](http://msdn.microsoft.com/en-us/library/office/microsoft.sharepoint.client.web.aspx) class |
| Site collection | A group of sites that share the same owner and administrative settings, such as permissions. |  | [Create a site collection in SharePoint 2013](http://technet.microsoft.com/en-us/library/cc263094.aspx) |
| Site provisioning | A process for equipping sites with the features, structure, brand, and other functionality. |  | [Site provisioning techniques and remote provisioning in SharePoint 2013](http://blogs.msdn.com/b/vesku/archive/2013/08/23/site-provisioning-techniques-and-remote-provisioning-in-sharepoint-2013.aspx) (Vesa Juvonen)  [Self-service site provisioning using apps for SharePoint 2013](http://blogs.msdn.com/b/richard_dizeregas_blog/archive/2013/04/04/self-service-site-provisioning-using-apps-for-sharepoint-2013.aspx) (Richard di Zerega)  [Site definitions and provisioning: the feature framework](http://msdn.microsoft.com/en-us/library/office/ms454453(v=office.12).aspx) (SharePoint 2010) |
| Subsite | A single SharePoint site in a SharePoint site collection. A subsite can inherit navigation and permissions from a parent site, or it can have unique permissions and navigation. |  | You can create subsites that are based on the root site collection, or sites based on other site collections. You can choose to inherit permissions from the site collection or specify unique permissions for the subsite. |

# Six Basic Things About SharePoint

To understand and use SharePoint branding and site provisioning capabilities, it’s important to first understand six basic things about SharePoint:

1. The hierarchy of SharePoint structural elements
2. Types of SharePoint sites and why you would use them
3. The file system and content databases, and how they work together
4. File customization states and their long-term impact on upgrades
5. Apps for SharePoint
6. Client-side programming in SharePoint with the .NET client-side object model (CSOM) and REST APIs

## The hierarchy of SharePoint structural elements

Conceptually, the hierarchy of SharePoint objects is expressed as containership: the types of objects at the type of the hierarchy contain all of the types of objects below them in the hierarchy.

**Table 3. The hierarchy of SharePoint structural elements**

|  |  |
| --- | --- |
| **Type of object (in order from top to bottom of the hierarchy)** | **Description** |
| Web applications | Web applications exist on a server and respond to requests from a browser. Web applications are not unique to SharePoint; they are the central structure in Internet Information Services (IIS).  In SharePoint, a web application is website with a unique URL and a separate content database stored in SQL Server. |
| Site collections | Containers of sites that define permissions, and that can define some aspects of branding, depending on the configuration, for all sites within the container. |
| Sites | A site is a collection of lists, libraries, structure, navigation, and look and feel elements usually organized around a central topic or theme.  Sites that are children of other sites in the same site collection are sometimes called subsites. A subsite is a site that is stored in a subfolder of the parent website. A subsite can inherit permissions and navigation structure from its parent site, or administration and authoring permissions may be unique for the subsite. Subsites can have child subsites. |
| Apps, lists, and document libraries | Containers of content and data that are organized into specific structures.  The Master Page Gallery is a special document library in SharePoint 2013 publishing sites where all branding elements—master pages, page layouts, JavaScript files, CSS, and images—are stored by default. Every site has its own Master Page Gallery.  In Team sites, the master page comes from the site, not the site collection. |
| Items | Individual pieces of content or data that are contained in apps, lists, and document libraries. |

## Types of SharePoint sites and why you would use them

You can create Internet, intranet, extranet, and personal (OneDrive for Business) sites with SharePoint 2013. Table 4 shows the types of features and site templates that support each type of site.

**Table 4. Types of SharePoint sites and why you should use them**

|  |  |  |  |
| --- | --- | --- | --- |
| **Site type** | **SharePoint site type** | **Site template title** | **Guidance** |
| Internet | Publishing | Publishing portal | Use for Internet sites. |
|  |  | Enterprise wiki | Use to publish information that you want to share across the enterprise. |
|  |  | Product catalog | Use to manage product catalogs for the enterprise. |
| Intranet | Collaboration | Team site | Use when your site needs to support content and document creation, information sharing, and collaboration. |
|  |  | Project site | Use to support project management and collaboration. |
|  |  | Community (wiki) site | Use to support community member exploration, content discovery, and topical discussion. |
|  | Publishing | Enterprise wiki | Use for large intranet portals. |
| Extranet |  | Publishing portal | For Internet sites. Users log on to extranet resources through the Publishing site. |
|  |  | (Other Collaboration site template types as needed to meet business objectives) |  |
| Personal | OneDrive for Business | MySiteHost | Hosts public profile pages and personal sites. |

To get all of the site templates for your SharePoint 2013 installation, use the [Get-SPWebTemplate](http://technet.microsoft.com/en-us/library/ff607910.aspx) PowerShell command.

## The file system and content databases, and how they work together

Understanding the SharePoint file system and content database is an important part of understanding branding options and upgrade and migration implications of site customizations.

### File system

SharePoint stores files in the file system (“hive”). In SharePoint 2013, this location is called the “15-hive”. The path to the “15-hive” is *%program files%/Common Files/Microsoft Shared/Web Server Extensions/15/*. The 15 hive includes several subfolders that store files you’ll use when branding and provisioning sites.

**Table 5. Hive subfolders and what they store**

|  |  |
| --- | --- |
| **Hive subfolder** | **Description** |
| Logs | Event and error logs |
| Master Page Gallery | Stores master pages, page layouts, and other branding files. |
| Resources | Contains files that referece resources in the SharePoint object model. |
| Templates | Stores features, site-level images, site templates that template the default SharePoint sites, and layouts. |

### Content databases

Content databases store SharePoint content objects, such as site collections.. A content database is automatically installed for every site collection when you deploy SharePoint 2013. All of the content for a site collection is stored in one content database on one server. However, a content database can be associated with more than one site collection, and you can [attach content databases to a SharePoint web application](http://technet.microsoft.com/en-us/library/cc825314.aspx). There may be times when you need to [move content from one content database to another](http://technet.microsoft.com/en-us/library/cc825328.aspx), such as when the size of the content will soon exceed the size of the content database.

Some characteristics of a content database vary depending on how the site collection is used. For example, sites are often write-intensive, while other types of content, such as read-only documents, are read-intensive. How content is used impacts aspects of the content database such as size and performance.

## File customization states and their effects on upgrade

The state of SharePoint files and content affects how easy it is to apply updates and controls whether SharePoint serves the file from the content database or the file system.By default, all SharePoint files are *uncustomized* and reside in matching states in the SharePoint file system and in the content database. When a file, a content database entry, or both are used in specific ways or changed, the state of that content may be affected.

**Table 6. File and content states**

|  |  |  |
| --- | --- | --- |
| **File or content state** | **Definition** | **Notes and guidance** |
| uncustomized | An attribute associated with a file that indicates that it hasn’t been modified. | More than one copy of a file can point to the same source. This makes it easier for designers to implement changes. |
| customized | An attribute associated with a file that indicates that it has been modified. | After a file becomes customized, it’s harder to apply broad updates.  Be very careful about what you customize. As a general rule, it’s better to use SharePoint’s default files and functionality than to customize system files or introduce customizations that need to be manually updated, which incurs long-term costs. |
| ghosted | .A file with a source that is stored outside the content database. | A pointer in the content database (the ghost of the file) still exists that tells SharePoint to look for the file’s source on the server’s file system. |
| unghosted | An uncustomized version of the source file resides in the content database. | Example: The SharePoint 2013 Design Manager feature creates a sandboxed solution to package branding files. It’s never added to the file system of the server, therefore by its definition its files are considered ghosted. But, the files it deploys are still in an uncustomized state. |

By default, all files that SharePoint installs are *uncustomized* and *ghosted*. If a file has been customized, it won’t be updated when updates are applied by new service packs, updates the SharePoint Online service, farm solutions, or sandboxed solutions.

## Apps for SharePoint

Apps for SharePoint are lightweight solutions that don’t install on the SharePoint host server, which means they don’t make excessive API calls to the host server. You can build apps for SharePoint using the cloud-app model (CAM). Users can discover and download apps from the Office Store or from the enterprise’s App Catalog. Learn more about apps for SharePoint in the [Apps for SharePoint overview](http://msdn.microsoft.com/en-us/library/office/fp179930.aspx).

The Branding and Site Provisioning solution pack introduces code and content that demonstrate and document using provider hosted apps for SharePoint and CAM to provision SharePoint site branding.

This content does not demonstrate or document creating Office Store apps.

## Client-side programming in SharePoint

Use client-side programming models to [build apps for SharePoint](http://msdn.microsoft.com/en-us/library/office/jj163230.aspx). The samples and examples in this solution pack are written in CSOM, which is the preferred programming model for SharePoint custom development, and can run on computers where SharePoint is not installed.

For the most part, CSOM is a set of wrappers on web service calls to the SharePoint .NET server object model.

Use the following resources to get started with SharePoint development with [Visual Studio](http://msdn.microsoft.com/en-us/library/ee330921.aspx) or [Napa](http://msdn.microsoft.com/en-us/library/office/jj220038.aspx), learn some CSOM coding fundamentals, and look up CSOM APIs that are commonly used.

If you’re new to developing apps for SharePoint, you can start to ramp up with the Hello World remote app sample ([CSOM](http://code.msdn.microsoft.com/sharepoint/SharePoint-2013-Hello-0fd15fbf), [REST](http://code.msdn.microsoft.com/sharepoint/SharePoint-2013-Hello-25f8c6f1)), learn [how to complete basic operations](http://msdn.microsoft.com/library/office/fp179912.aspx) with client library code, and perform basic data access operations in apps ([CSOM](http://code.msdn.microsoft.com/sharepoint/SharePoint-2013-Perform-eba8df54), [REST](http://code.msdn.microsoft.com/sharepoint/SharePoint-2013-Perform-335d925b)). If you prefer developer training materials, see the [apps for Office and apps for SharePoint developer training](http://msdn.microsoft.com/en-us/office/aa905340), and additional MSDN resources for [developing apps for SharePoint](http://msdn.microsoft.com/en-us/library/office/jj163794.aspx).

Other modules in the solution pack refer to code samples and include example code that you can use to get started with the site provisioning and branding customization with apps for SharePoint.

If you’re interested in even more apps for SharePoint samples, consider downloading the [apps for SharePoint sample pack](http://code.msdn.microsoft.com/sharepoint/Apps-for-SharePoint-sample-64c80184).

[SharePoint 2013 .NET Server, CSOM, JSOM, and REST APIs](http://msdn.microsoft.com/en-us/library/office/apps/dn268594.aspx)

[.NET client API reference for SharePoint 2013](http://msdn.microsoft.com/en-us/library/jj193041.aspx)

[Apps for SharePoint references](http://msdn.microsoft.com/en-us/library/office/jj220040.aspx)

[Get Started with the SharePoint 2013 REST service](http://msdn.microsoft.com/en-us/library/office/apps/fp142380.aspx)

# The scenario: Site branding and provisioning with the Cloud App Model (CAM)

In SharePoint 2013, you can use custom CSOM code in apps for SharePoint to provision SharePoint site collections, sites, and subsites with branding elements. This site provisioning pattern is called [*remote provisioning*](http://blogs.msdn.com/b/vesku/archive/2013/08/23/site-provisioning-techniques-and-remote-provisioning-in-sharepoint-2013.aspx). SharePoint is increasingly focused on cloud-based deployments, so this pattern was created to help you use SharePoint’s out-of-the-box capabilities to provision site branding in way that reduces complexity and long-term operational costs.

## What can I do with the cloud-app model?

Sometimes, there is no correlation between features in full-trust code (FTC) and cloud app model (CAM). When developing a customization based on apps for SharePoint and CAM, consider an alternative approach rather than a direct conversion, and strive to keep customizations as simple as possible. Here are some examples:

* Replace event receivers with [remote event receivers](http://msdn.microsoft.com/en-us/library/office/jj220043.aspx).
* Replace site templates, web templates, and site definitions with remote provisioning. This works for both subsites and site collections.
* Replace timer jobs with Windows Azure or on-premises [worker roles](http://msdn.microsoft.com/en-us/library/windowsazure/jj149831.aspx).

Some things, such as HTTP modules and HTTP handlers can not be build with CAM. Before trying to replicate an existing customization in CAM, first understand why these customizations were built and if an out-of-the-box SharePoint feature can work.

## The remote provisioning pattern

Remote provisioning uses new app patterns to move provisioning logic outside of the SharePoint farm entirely. This approach eliminates the need to use the [feature framework](http://msdn.microsoft.com/en-us/library/office/ms454453(v=office.12).aspx) or other customizations in the SharePoint farm, and instead control customizations outside of SharePoint. This approach makes it possible to update and change the provisioning engine without impacting the availability of SharePoint.

While aspects and implementations of the remote provisioning pattern is documented in detail throughout the solution pack, you may find it useful to get started with the following introductions to the pattern.

[Self-service site provisioning using apps for SharePoint 2013](http://blogs.msdn.com/b/richard_dizeregas_blog/archive/2013/04/04/self-service-site-provisioning-using-apps-for-sharepoint-2013.aspx) (Richard diZerega)

[SharePoint 2013 site provisioning techniques video recording](http://blogs.msdn.com/b/vesku/archive/2013/09/09/sharepoint-2013-site-provisioning-techniques-presentation-video-recording.aspx) (Vesa Juvonen)

[Why did we introduce the SharePoint app pattern?](http://blogs.msdn.com/b/vesku/archive/2013/09/06/introduction-to-sp-apps-or-cam.aspx) (Vesa Juvonen)

### Remote provisioning 101

In the simplest implementation of the remote provisioning pattern, provisioning requirements are stored in a SQL or SQLAzure database or XML file; and an app for SharePoint reads requirements from the data source, reads branding elements from their source location, and applies branding elements to the target site based on the provisioning requirements.

The solution pack follows this sequence of events to demonstrate the remote provisioning pattern.

**Table 7. Basic remote provisioning sequence and their associated samples and module documents**

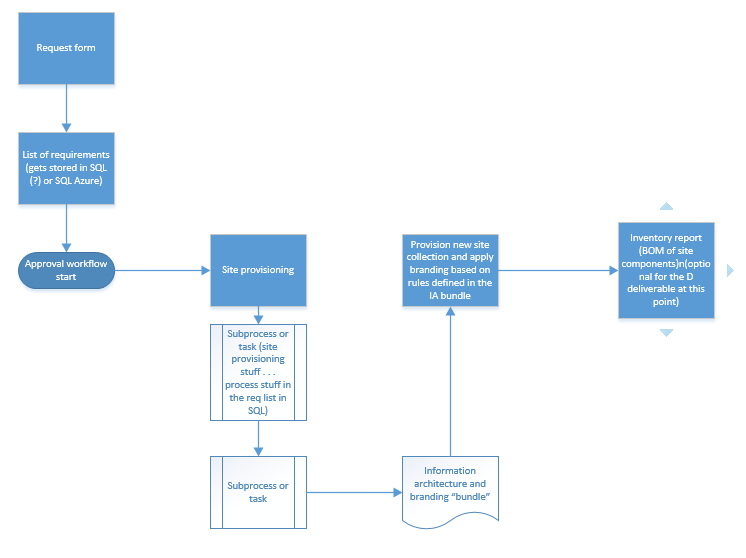
|  |  |  |  |
| --- | --- | --- | --- |
| **Step** | **Description** | **Sample(s)** | **Modules** |
| 1 | The user requests a change to the site through a form, which kicks off an approval workflow.  The data that the user submits via the request form are stored using potentially any data storage format (e.g., SQL, SQL Azure, XML). | [Self-service site provisioning using apps for SharePoint 2013](http://blogs.msdn.com/b/richard_dizeregas_blog/archive/2013/04/04/self-service-site-provisioning-using-apps-for-sharepoint-2013.aspx) (Richard diZerega)  [SiteProvisioningWorkflow](http://code.msdn.microsoft.com/SharePoint-2013-Use-e2ee88dd)  [SiteProvisioningWorklowAppWeb](http://code.msdn.microsoft.com/SharePoint-2013-Use-2b96feb7) | Module 5 |
| 2 | If the workflow is approved, the app for SharePoint calls the stored data and provisions the site according to the metadata that user submitted in his or her request in step 1. | [Batch Provisioning](http://code.msdn.microsoft.com/Provision-sites-in-batches-fcf31bc6)  [SiteProvisioningWorkflow](http://code.msdn.microsoft.com/SharePoint-2013-Use-e2ee88dd)  [SiteProvisioningWorkflowAppWeb](http://code.msdn.microsoft.com/SharePoint-2013-Use-2b96feb7)  [ServicesSiteManager](http://code.msdn.microsoft.com/SharePoint-2013-Use-apps-9094e012) | Module 5 |
| 3 | The app for SharePoint scopes provisioning to the instructions in the request form by using the data available in the app web and content database. During this stage, applicable branding elements are provisioned to the site. | [AlternateCSSAppAutohosted](http://code.msdn.microsoft.com/SharePoint-2013-Use-an-app-c5d78f25)  [ApplyThemeApp](http://code.msdn.microsoft.com/SharePoint-2013-Use-an-app-d90a49e3)  [OneDriveForBusinessSiteBranding](http://code.msdn.microsoft.com/SharePoint-2013-Brand-a-6da627cb)  [ProvisionCustomCSS](http://code.msdn.microsoft.com/SharePoint-2013-Provision-bf1d878a)  [ProvisionWikiPages](http://code.msdn.microsoft.com/SharePoint-2013-Use-an-app-5db977e8) | Module 2  Module 4  Module 5 |

**Note** All links in Table 7 point to individual samples.

**Note** While Table 7 presents five steps that may be typical of a remote provisioning scenario, the samples you would use depend on the approach that works best for your enterprise. For example, if there is no justifiable business need to create a custom approval workflow, it’s OK not to use that sample. The scenario is one potentially common use-case.

Figure 2 illustrates the scenario.

**Figure 2. Example of a site provisioning and branding flow powered by the remote provisioning pattern**



### Solution pack samples

Samples demonstrate the core scenario and extend it to cover some more specific use cases. Additionally, the modules contain example code not bundled as standalone samples. Table 8 lists samples that specifically demonstrate site provisioning functionality, white Table 9 lists samples that demonstrate how to use the SharePoint CSOM to complete common branding tasks.

**Table 8. Site provisioning samples**

|  |  |  |
| --- | --- | --- |
| **Sample name** | **Description** | **Associated module** |
| [Batch Provisioning](http://code.msdn.microsoft.com/Provision-sites-in-batches-fcf31bc6) | Provisions site collections in a console app. | Module 5 |
| [ProvisionWikiPages](http://code.msdn.microsoft.com/SharePoint-2013-Use-an-app-5db977e8) | Demonstrates how to use the remote provisioning model to provision a Wiki page and add and remote Web Parts and HTML from the Wiki page. | Module 2 |
| [SiteProvisioningWorkflow](http://code.msdn.microsoft.com/SharePoint-2013-Use-e2ee88dd) | Provisions site collections with a workflow on the host web and a remote event receiver: Site collection provisioning in a remote event receiver, with a workflow deployed to the host web. | Module 5 |
| [SiteProvisioningWorkflowAppWeb](http://code.msdn.microsoft.com/SharePoint-2013-Use-2b96feb7) | Provisions site collections with a workflow on the app web and a remote event receiver: approval workflow on the app web  with site collection provisioning in a remote event receiver. | Module 5 |

**Note** The **BatchProvisioning**, **SiteProvisioningWorkflow**, and **SiteProvisioningWorkflowAppWeb** samples demonstrate the core concepts and functions of the remote provisioning pattern. The **ProvisionWikiPages** sample addresses a specific use case (Wiki page provisioning).

**Table 9. Branding samples**

|  |  |  |
| --- | --- | --- |
| **Sample name** | **Description** | **Associated module** |
| [AlternateCssAppAutohosted](http://code.msdn.microsoft.com/SharePoint-2013-Use-an-app-c5d78f25) | Uses an auto-hosted app for SharePoint to demonstrate how to use an app-installed event handler to brand a site with CSOM. | Module 4 |
| [ApplyThemeApp](http://code.msdn.microsoft.com/SharePoint-2013-Use-an-app-d90a49e3) | Demonstrates how to apply a theme (CSOM). | Module 4 |
| [OneDriveForBusinessSiteBranding](http://code.msdn.microsoft.com/SharePoint-2013-Brand-a-6da627cb) | Demonstrates how to use the remote provisioning model to provision a Wiki page and add and remote Web Parts and HTML from the Wiki page. | Module 4 |
| [ProvisionCustomCSS](http://code.msdn.microsoft.com/SharePoint-2013-Provision-bf1d878a) | Demonstrates how to set custom CSS to the host site using a user customer action and JavaScript injection pattern (CSOM). | Module 4 |
| [ServicesSiteManager](http://code.msdn.microsoft.com/SharePoint-2013-Use-apps-9094e012) | Demonstrates how to use a service to encapsulate all of the information in SharePoint host web to the app web, and get a list of site collections in a specified web application and create a content type with a specific **ContentTypeId**. This sample is especially useful when you want to use the remote provisioning pattern to provision sites using apps for SharePoint, but the CSOM member you need to complete your scenario is not yet available in CSOM. | Module 5 |

In addition to standalone samples, solution pack modules include inline code that demonstrate related scenarios and programming approaches. Table 10 provides an index of code found and referenced in the module documents.

**Table 10. Inline code index**

|  |  |
| --- | --- |
| **Code example** | **Module** |
| DesignPackage code example | Module 3 |
| FileCreationInformation code example | Module 3 |
| ApplyTheme code example | Module 4 |
| Use custom actions to provision branding elements | Module 4 |
| Add a site logo | Module 4 |
| OneDrive for Business site branding excerpt | Module 4 |
| Use TenantAdmin CSOM to create a site collection on SharePoint Online | Module 5 |
| Deploy a custom WCF solution to a SharePoint farm | Module 5 |

## Remote provisioning impacts pre-existing site content

Depending on the specific site elements you want to provision, your code will override default or pre-existing site content with a hook for the remote provisioning app for SharePoint. The app will select site templates and other capabilities based on the provisioning requirements stored in the database, without configuring SharePoint at all.

The basic remote provisioning pattern is the same regardless of additional requirements. However, when planning to use this pattern to provision site branding, map your brand development strategy in the context of the customization capabilities that SharePoint CSOM, JSOM, and REST APIs provide (all sample code in the solution pack uses CSOM). Also consider:

* Site architecture: are you building an Internet-facing site, intranet site for collaboration or other function used internally, or an extranet that requires authorized users to log on through the Internet-facing to site access company data?
* The degree of control that specific users have to define and request provisioning requirements. Should users be able to specify custom provisioning options using a form? Are changes applied to the site automatically, only after people with decision-making power approve the changes, or are they managed by a governance policy?
* The types of branding customizations you want to apply (structural, look and feel, or both)?

# A brief overview of the SharePoint branding workflow

Branding a SharePoint website is a lot like branding other websites: you use web technologies that you’re familiar with, such as HTML, CSS, and JavaScript to build the structure, look and feel, and custom behavior of your sites. SharePoint is also based on ASP.NET, and uses a page model that is very similar to the [ASP.NET master page/page layout model](http://msdn.microsoft.com/en-us/library/aa479007.aspx). The page model encompasses the structure and provides hooks and logic for applying look and feel elements.

SharePoint provides numerous Web Parts you can use to incorporate data views, images, scripts, search results, and more into your site design. Composed looks provide an easy way for users to customize the look and feel of their site while reinforcing designer and IT department control over design details and look and feel options that are available, and both the theming engine and custom CSS capabilities open the door for more advanced branding customizations.

## High-level view of the branding workflow

The branding design and development workflow for SharePoint websites closely resembles the design workflow the industry uses:

1. Plan your site architecture and design.
2. Create design assets using familiar web design tools and technologies.
3. Build your site using SharePoint tools such as Design Manager.
4. Package your site design, and use apps for SharePoint and the remote provisioning pattern to provision site branding.

**Note** Applying branding in SharePoint means modifying the look and feel of a default SharePoint site. This can include making both structural and cosmetic changes to the site’s appearance.

## Branding cost and complexity

Branding changes range from low-cost and simple to high-cost and complex. Through the UI, users can apply composed looks, which include a background image, color palette, fonts, and a master page associated with these elements, and a preview file associated with the master page. You can use the SharePoint 2013 theming engine to create your own themes, and you can create custom CSS to modify the look and feel of your site.

**Important**  While it’s possible to create custom master pages and other structural elements as part of a custom branding project, the long-term cost of supporting structural customizations can be high, and may make it more costly for your organization to apply upgrades and support the long-term applicability of short-term investments in customization.

## Branding SharePoint sites hosted on-premises or on a Dedicated farm

You can use the remote provisioning pattern to brand Team sites, Publishing sites, and OneDrive for Business sites that are hosted on-premises or on a Dedicated farm at both the site collection and subsite level.

## SharePoint Online

Part of planning a SharePoint branding project is deciding which types of site(s) you want to build, brand, and provision. SharePoint Online licensing impacts whether publishing site capabilities are available to you. While all licenses enable you to specify at least one public website that has some of the features of a SharePoint Server Publishing site, not all licenses provide full Publishing site capabilities.

**Table 9. Site options in SharePoint Online**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Office 365 edition** | **Team site?** | **Public website?** | **Publishing site?** | **Notes** |
| Small Business | x | x |  | Includes one Team site and the public website. Does not include Publishing site functionality. The public website capabilities were designed with small business in mind. |
| Enterprise | X |  | X | Includes a Team site collection at the root web application for the domain that does not include Publishing, and you can create new Publishing site collections under that root web application. |

[Select an Office 365 plan for business](http://office.microsoft.com/en-us/business/compare-office-365-for-business-plans-FX102918419.aspx)

[Model: Design and branding in SharePoint 2013](http://www.microsoft.com/en-us/download/details.aspx?id=30715)

# When should I customize?

Most functionality you need to meet your business needs in available out-of-the-box in SharePoint.

Before creating a customization, determine whether there is an actual business case for creating the customization and what the long-term costs of creating and supporting this customization would be to the enterprise. How are features and functionality provided for end-users? Look at business goals and user experience considerations before technology.

When working with an existing custom SharePoint solution and weighing whether and how to migrate it to the cloud-app model, first understand why the customization was done and what purpose it serves.

When considering moving an existing customization from full-trust code (FTC) to the cloud app model (CAM), there usually isn’t a one-to-one relationship between FTC features and functionality, and CAM. Rather than trying to find a one-to-one match between server-side and client-side code, consider alternative approaches. Table 10 maps commonly-used some concepts and functionality of SharePoint solutions to their equivalents in apps for SharePoint.

**Table 10. Mapping SharePoint concepts to apps**

|  |  |  |  |
| --- | --- | --- | --- |
| **I need to . . .** | **In a SharePoint solution I used . . .** | **In apps for SharePoint, I use . . .** | **Notes** |
| display information in SharePoint pages. | Web Parts. | App Parts. | Web parts run on the SharePoint server with user permission or full-permissions/elevated privilege.  App parts run in the browser or on an external server with an app identity with specifically granted permissions. They are completely isolated on the client in their own domain. App parts are executed outside of SharePoint and incur no performance impact on the SharePoint server.  [How to: Create app parts to install with your app for SharePoint](http://msdn.microsoft.com/en-us/library/office/fp179921(v=office.15).aspx) |
| create and manage notifications. | event receivers and feature receivers. | Remote event receivers and app event receivers. | Event receivers and feature receivers require server-side code and can’t notify external systems of events.  Remote event receivers use client-side code, can be used in SharePoint solutions or apps for SharePoint, and can notify external systems of events.  App event receivers execute code when apps are installed, uninstalled, or upgraded.  [Handling events in apps for SharePoint](http://msdn.microsoft.com/en-us/library/office/jj220048%28v=office.15%29.aspx)  [How to: Create an event receiver for an app for SharePoint](http://msdn.microsoft.com/en-us/library/office/jj220051%28v=office.15%29.aspx) |
| access data. | the .NET server object model (SSOM), .NET client object model (CSOM), and OData. | .NET client object model (CSOM, JSOM), OData, REST, cross-domain libraries. | [How to: Complete basic operations using SharePoint 2013 client library code](http://msdn.microsoft.com/en-us/library/office/fp179912(v=office.15).aspx)  [How to: Complete basic operations using JavaScript library code in SharePoint 2013](http://msdn.microsoft.com/en-us/library/office/jj163201(v=office.15).aspx)  [Get started with the SharePoint 2013 REST service](http://msdn.microsoft.com/en-us/library/office/fp142380(v=office.15).aspx)  [.NET client API reference for apps for SharePoint](http://msdn.microsoft.com/en-us/library/office/jj220029(v=office.15).aspx)  [JavaScript API reference for apps for SharePoint](http://msdn.microsoft.com/en-us/library/office/jj220031(v=office.15).aspx)  [REST API reference and samples](http://msdn.microsoft.com/en-us/library/office/jj860569(v=office.15).aspx) |
| package and deploy. | solution packages (WSPs, feature packages). | app catalog and Office Store. | Solution packages are difficult to deploy across a SharePoint farm.  You can publish an app for SharePoint to the Office Store if you want to make it publicly available or sell it.  Use the app catalog to make an app for SharePoint available within your organization.  Guidance and code samples in the solution pack demonstrate how to use apps for SharePoint to provision branding elements to your SharePoint site.  [How to: Set up an app catalog on SharePoint Online](http://msdn.microsoft.com/en-us/library/office/dn574752(v=office.15).aspx)  [How to: Set up an app catalog on SharePoint](http://msdn.microsoft.com/en-us/library/office/fp123530(v=office.15).aspx)  [Publish apps for Office and SharePoint to the Office Store](http://msdn.microsoft.com/en-us/library/office/jj220037(v=office.15).aspx)  [Choose patterns for developing and hosting your app for SharePoint](http://msdn.microsoft.com/en-us/library/office/fp179887%28v=office.15%29.aspx) |
| use external data. | external content types. | app-scoped external content types. | SharePoint site administrators or SharePoint Designer users must create and/or install external content types, which, can be installed only at the farm level.  App-scoped external content types apply only to the app for SharePoint for which they were created, require no administration, and can access OData sources.  [App-scoped external content types in SharePoint 2013](http://msdn.microsoft.com/en-us/library/office/jj163918%28v=office.15%29.aspx)  [How to: Create an external content type from OData sources in SharePoint 2013](http://msdn.microsoft.com/en-us/library/office/jj163967.aspx) |
| add custom pages and master pages. | application pages and site pages. | Web hosted pages | Application pages are shared across all istes on the server and are hosted on SharePoint.  Site pages are hosted by SharePoint, and require that page controls be listed in a safe controls list.  While application pages are ideal for custom code, custom code on site pages will break after customization.  Instead, use Web hosted pages. They are designed to be customizable, support the use of built-in Web Parts on site pages, are hosted externally, and are available anywhere the app is installed. |

The bottom line? To help maintain costs and reduce complexity, design for simplicity.

Want to see some of the information in this table as an infographic? [Get started with apps for SharePoint and see the big picture](http://msdn.microsoft.com/en-US/office/dn448479).

# Conclusion

Module 1 introduced the solution pack concept, six basic things about SharePoint, the remote provisioning pattern presented in the solution pack modules and code samples, high-level things to consider when planning a SharePoint site branding project, and SharePoint Online site options.

Next, Module 2 introduces the SharePoint 2013 page model, followed by Module 3, which explains tools you can use to manage branding assets for SharePoint and the Design Packages API.

# Updates

August 7, 2014:

Added **Development and Integration SKUs** section (Section 3) to Module 3.

Added link to new asynchronous site provisioning code sample -- SharePoint 2013: Provision dedicated and on-premises sites with the app model – to Module 5.

Added link to sideloading sample – Use sideloading to provision a provider-hosted app – to Modules 5 and 6.

Added link to page manipulation sample – Use an app for SharePoint to provision a wiki page-- to Module 6.