**SharePoint Online Site Template Scripts Research**

**Primary References**

[The Ultimate Guide to SharePoint Site Templates and Site Scripts (laurakokkarinen.com)](https://laurakokkarinen.com/the-ultimate-guide-to-sharepoint-site-designs-and-site-scripts/#what-are-site-designs-and-site-scripts)

[SharePoint site template and site script overview | Microsoft Learn](https://learn.microsoft.com/en-us/sharepoint/dev/declarative-customization/site-design-overview)

[SharePoint site design - PowerShell cmdlets | Microsoft Learn](https://learn.microsoft.com/en-us/sharepoint/dev/declarative-customization/site-design-powershell)

[sp-dev-site-scripts/samples at master · pnp/sp-dev-site-scripts (github.com)](https://github.com/pnp/sp-dev-site-scripts/tree/master/samples)

[Site template JSON schema | Microsoft Learn](https://learn.microsoft.com/en-us/sharepoint/dev/declarative-customization/site-design-json-schema)

[Deploy Managed Metadata fields using site designs and site scripts – Beau Cameron](https://beaucameron.com/2018/11/06/deploy-managed-metadata-fields-using-site-designs-and-site-scripts/)

[PnP provisioning framework | Microsoft Learn](https://learn.microsoft.com/en-us/sharepoint/dev/solution-guidance/pnp-provisioning-framework)

**Site Templates Overview**

A site script is a JSON string that contains the actions to be done on a site. A site template, on the other hand, is like a container for the site script. You can attach one or more site scripts to a site template, and when you apply the site template, you execute the actions defined in all of the site scripts. It’s like the site template is the visible surface part that can group several site scripts together, and the site scripts are all the action underneath.

Site templates are more like scripts (they consist of site scripts after all) that you run to configure a site automatically.

An end user can select your custom site template when they are creating a new modern site in SharePoint, performing the actions right there and then. You can also apply site templates to existing sites programmatically by using CSOM, PowerShell or REST.

**Start a Flow or Logic App**

Being able to start a Flow or a Logic App by applying a site template opens a massive amount of possibilities. You can, for example, integrate it with an Azure Function that applies a PnP provisioning template on the site, or even make calls to Microsoft Graph.

**Tools for Designing Site Templates**

In addition to constructing the JSON by hand, you can also use third-party tools with graphical user interfaces that generate the site scripts for you, such as [sitedesigner.io](https://www.sitedesigner.io/) by Mikko Punamäki or [Site Template Studio](https://github.com/SharePoint/sp-dev-solutions/tree/master/solutions/SiteDesignsStudio) by Yannick Plenevaux. Both of these methods of constructing site scripts have their pros and cons.

When it comes to the two tools I just mentioned:

* The [sitedesigner.io](https://www.sitedesigner.io/) web site is great if you just want to quickly generate your site script. You can open it in your browser and get to work immediately.
* The [Site Template Studio](https://github.com/SharePoint/sp-dev-solutions/tree/master/solutions/SiteDesignsStudio) is an SPFx web part that you first need to install to your SharePoint Online tenant. However, the benefit of the web part is that you can also deploy site templates through it! No need to run a separate PowerShell script.

**Export List Configurations**

If you are familiar with PnP provisioning, you have most likely configured sites and then exported them as templates. With site templates, you can export an existing list configuration as a site script in a similar manner. You can do it with the following script. The Get-SPOSiteScriptFromList commandlet prints out the site script to the console, and you can copy it from there to a file of your choice.

**Export Site Configurations**

The ability to export entire site configurations was announced at the SharePoint Conference in May 2019. With the Get-SPOSiteScriptFromWeb commandlet you can export the settings of your choosing as well as the configurations for lists/libraries at the same time; no need to export lists with their own command separately (make sure to use their internal names).

Note that the command will not export page layout and web part configurations — those features are not yet available in site templates, although they have been mentioned to be implemented sometime in the future.

**Site Templates Limitations**

The basic **createSiteColumn** action supports only the following types of columns: Text, Note, Number, Boolean, User, and DateTime.

Currently site templates cannot, for example, adjust page layouts and web part settings. However, there is a thing called [PnP provisioning engine](https://docs.microsoft.com/en-us/sharepoint/dev/solution-guidance/pnp-provisioning-framework) (which I’ve already mentioned a couple of times in this blog post) that does a similar thing as site templates but is more feature rich. It is very likely, that you’ll find what you need in its [latest schema](https://github.com/SharePoint/PnP-Provisioning-Schema).

**Deploying Site Templates**

You can deploy your site script and attach it to a new site template with the PowerShell script below. Before running the script, change the variable values to fit your purpose. Is your site script for a team or a communication site? What would be a descriptive name and description for it? Do you want to use a custom preview image or the default one? The script below expects the JSON file to be in the same folder with it. However, you can change the path to point somewhere else too if you like.

$adminSiteUrl = "https://mytenant-admin.sharepoint.com"

$siteScriptFile = $PSScriptRoot + "\CustomTeamSiteScript.json"

$webTemplate = "64" #64 = Team Site, 68 = Communication Site, 1 = Groupless Team Site

$siteScriptTitle = "Custom Team Site Script"

$siteDesignTitle = "Custom Team Site Template"

$siteDesignDescription = "Custom team site template with multi-colored theme, external sharing disabled and some cool stuff via Power Automate."

$previewImageUrl = "/SiteDesignPreviewImages/custom-team-site-design-preview-image.png" # if left empty, the default preview image will be used.

$designPackageId = "6142d2a0-63a5-4ba0-aede-d9fefca2c767" # The default site template to use as a base when creating a communication site, more info later.

$cred = Get-Credential

Connect-SPOService $adminSiteUrl -Credential $cred

$siteScript = (Get-Content $siteScriptFile -Raw | Add-SPOSiteScript -Title $siteScriptTitle) | Select -First 1 Id

Add-SPOSiteDesign -SiteScripts $siteScript.Id -Title $siteDesignTitle -WebTemplate $webTemplate -Description $siteDesignDescription -PreviewImageUrl $previewImageUrl -DesignPackageId $designPackageId

**Maintaining Site Templates**

One important thing to be aware of is that you can attach multiple site scripts to one site template. Why is this important? Think about a situation where you have many site templates, and all of them include a set of certain actions. For example, all the site templates apply the same company branded color theme and set the same regional settings but have different list configurations. As time goes by, you might need to make changes to these actions. It would end up being a lot of work to update each of the site templates individually.

Instead, you can have one or more “common” site scripts, where you’ll put the actions that are shared by your site templates. Then you include both this common site script and the site template specific site script in that single site template. This way, you’ll only need to modify one site script when you want to make changes to the actions shared by all your site templates.

**Edit and Delete**

Edit a Deployed Site Template – You can edit a deployed site template by using the *Set-SPOSiteDesign* commandlet.

Edit an Existing Site Script – You can edit a deployed site template script by using the *Set-SPOSiteScript* commandlet

Delete an Existing Site Template – If you ever want to delete any of your site templates or scripts, you can do that with these commands. Deleting a site template doesn’t automatically remove the site script attached to it, you need to do that separately. Remember though, that another site template might use the same site script, in which case you should not delete it.

**Applying Templates**

You can also apply site templates to a site via the UI after the site has been created. This happens through the new site template panel which you can find behind the cogwheel → Site templates option if you are a site owner or admin, and your tenant is in targeted release.

**Site Templates Security**

Limit Access:

All the site templates are public and visible to everyone by default (unless group creation has been limited in the tenant — check the “The Web Template” property section for more info). After you grant permissions to certain groups or users, only those will be able to see and use the site template. Limiting the visibility of the default site templates is not possible.

You can set the site template permissions with the script below. You can assign permissions to security groups as well as individual users. If you want to assign permissions to several principals, separate them with a comma. The value of the Rights parameter is always View.

$adminSiteUrl = "https://mytenant-admin.sharepoint.com"

$siteDesignId = "eb13d8b6-c835-4b07-80fd-d01bd10de666"

$principals = "Security Group Name", "user@mytenant.onmicrosoft.com"

$cred = Get-Credential

Connect-SPOService $adminSiteUrl -Credential $cred

Grant-SPOSiteDesignRights -Identity $siteDesignId -Principals $principals -Rights View

Get-SPOSiteDesignRights -Identity $siteDesignId

Revoke Limits:

Later, if you want to remove the permissions, use the Revoke-SPOSiteDesignRights commandlet and specify the principals you want to remove permissions from. If you remove all permissions, the site template will become public again.

$adminSiteUrl = "https://mytenant-admin.sharepoint.com"

$siteDesignId = "eb13d8b6-c835-4b07-80fd-d01bd10de666"

$principals = "user@mytenant.onmicrosoft.com"

$cred = Get-Credential

Connect-SPOService $adminSiteUrl -Credential $cred

Revoke-SPOSiteDesignRights -Identity $siteDesignId -Principals $principals

Get-SPOSiteDesignRights -Identity $siteDesignId

**Apply Site Templates Programmatically:**

You can apply your site template on a site with the script below. The GUID for the Identity parameter can be printed out with the Get-SPOSiteDesign command that was presented earlier in this post. If your site template has more than 30 verbs in total, you need to use the Add-SPOSiteDesignTask commandlet instead. See the [Large site templates](https://laurakokkarinen.com/the-ultimate-guide-to-sharepoint-site-designs-and-site-scripts/#large-site-designs) section for more information.

$adminSiteUrl = "https://mytenant-admin.sharepoint.com"

$siteUrl = "https://mytenant.sharepoint.com/sites/siteUrl"

$siteDesignId = "eb13d8b6-c835-4b07-80fd-d01bd10de666"

$cred = Get-Credential

Connect-SPOService $adminSiteUrl -Credential $cred

Invoke-SPOSiteDesign -Identity $siteDesignId -WebUrl $siteUrl

**Associating a Site Template to a Hub Site:**

These days you can also associate a site template to a hub site. What this means is that when you join a site to the hub, the site template will automatically get applied to the joined site.

The site template WebTemplate value doesn’t necessarily need to match the site type you are planning on joining the hub. As mentioned in [The WebTemplate property](https://laurakokkarinen.com/the-ultimate-guide-to-sharepoint-site-designs-and-site-scripts/#the-web-template-property) section, the property value only controls in which dropdown your site template appears in the Create site UI.

You can join both team sites and communication sites to the same hub, and the site template will get executed successfully for both. If some actions in the site template are not compatible with the site type, those actions will either be skipped, or they will fail gracefully.

Once you have a site template attached to a hub site, it will get executed to all sites that are added to the hub.

**The Future of Site Templates**

New actions are continuously being added to the SharePoint site template schema. Microsoft wants site templates to be able to do everything PnP provisioning templates can do at some point [[source](https://youtu.be/FlfKmwYl7kc?t=39m56s)]. However, due to the limited amount of resources, the development of site templates is much slower than the development of the community-driven PnP provisioning templates. That’s why we are still using PnP provisioning until site templates have time to catch up. Whether that will actually ever happen is up for debate.

There are still a lot of things you can do with PnP provisioning that you can’t do with SharePoint site templates. But then again, there are also things you can’t do with PnP provisioning that you can do with site templates, such as restrict external sharing and start a Power Automate flow. So, now, site templates and PnP provisioning templates complement each other, and that’s why you’ll most likely find yourself using them both.