SPSync

Admin permissions pack (PDF)

The Microsoft Graph permissions SPSync asks for, the admin consent flow, the per-site and per-list grant commands, and what SPSync never requests.

This page is for the person who receives the request. Someone in finance or operations wants a SharePoint list copied into a database, and the consent prompt has arrived with the Entra or SharePoint admin. Most admins want two answers before they click anything, namely how narrow the grant can be and whether the app can change SharePoint. The short answers are one list at a time, and no. Forward the PDF to whoever signs off security reviews.

Delegated or application access

SPSync reads SharePoint through Microsoft Graph and nothing else. It can run as the person who set it up (delegated) or as the app itself (application).

	Delegated	Application
Runs as	The signed-in user, with that user's own access	The SPSync app, unattended
Who consents	The user, for themselves	A tenant admin, then a grant per site or list
Shortest sync interval	5 minutes	1 minute (30 seconds absolute minimum)
Keeps running when the user is away	No, it stops when the refresh token expires	Yes
Use it for	Trials and small-team use	Scheduled production replications

Delegated sign-in asks for Sites.Read.All, User.Read and offline_access. Because it is delegated, Sites.Read.All gives SPSync the signed-in user's own read access and no more than that. Any user can consent for themselves unless your tenant requires admin consent for every app.

Application permissions, narrowest first

Unattended sync needs application permissions, and this is the part that needs an admin. SPSync offers three, and we recommend the first one that fits.

Permission	Scope	Notes
Lists.SelectedOperations.Selected	Only the lists an admin names	Tightest. A grant on one list breaks that list's permission inheritance from its site
Sites.Selected	Only the sites an admin names	Every list and library on a granted site, no others. Inheritance is untouched
Sites.Read.All	Every site in the tenant, read only	Use only when per-site grants are unworkable, for example when the sites to read change too often to grant one by one

With either Selected permission, consent on its own gives SPSync access to nothing. Access starts only when an admin grants a named site or list, and every grant SPSync requests is read only.

The list-level grant is newer than Sites.Selected and less familiar. Microsoft's overview of Selected permissions describes Lists.SelectedOperations.Selected as application access to a specific list, and notes that assigning an application to a list breaks inheritance on that list, so keep an eye on the service limits for unique permissions if you grant many lists. A grant on the site does not break inheritance, because the site is where inheritance starts. The same page lists item, folder and file scopes as well; SPSync uses none of them, since it replicates whole lists.

The admin consent flow

The first step is consent on the app registration. A Global Administrator or Privileged Role Administrator opens this link once, signed in to your tenant:

```
https://login.microsoftonline.com/{tenant-id}/adminconsent?client_id={client-id}
```

For Sites.Read.All that link is the whole grant. For the two Selected permissions, follow it with one grant per site or per list. SPSync's setup screen does the grant for you only when the connection's own app holds Sites.FullControl.All (a bring-your-own app can; a user sign-in never does). Otherwise it prints the command for your admin to run straight away, with a line saying why, and after a grant it reads the stored role back and corrects it if Graph stored a different one. For a site, that is PnP PowerShell:

```
Grant-PnPEntraIDAppSitePermission -AppId "{client-id}" -DisplayName "{app-name}" -Site
https://contoso.sharepoint.com/sites/finance -Permissions Read
```

The public Connect to SharePoint page has the app registration and certificate steps that come before these grants.

PnP has no cmdlet for a list-scoped grant, so the list form calls Graph directly through the Microsoft Graph PowerShell SDK:

```
Connect-MgGraph -Scopes Sites.FullControl.All
Invoke-MgGraphRequest -Method POST -Uri
"https://graph.microsoft.com/v1.0/sites/{site-id}/lists/{list-id}/permissions" -Body (@{
roles = @("read"); grantedToIdentities = @(@{ application = @{ id = "{client-id}";
displayName = "{app-name}" } }) } | ConvertTo-Json -Depth 5)
```

To check what has been granted on a site, run Get-PnPAzureADAppSitePermission. Each grant can be removed at any time from the same tools, per site or per list, without contacting us.

What SPSync never asks for

Every grant is read. SPSync holds no write permission on SharePoint and has no code path that writes to it, so lists, libraries and permissions stay as they are. Nothing is installed in SharePoint either, so there is no add-in or site script to review.

Apart from User.Read for the signed-in user's own profile, SPSync requests no Graph permission outside SharePoint sites. It does not read mail or OneDrive files, and it does not need

Sites.FullControl.All for itself. Item-level permissions aren't read during a sync either, so the replica carries rows without SharePoint's sharing settings. Decide who sees which rows with database roles and row-level security on the destination. On a granted site, the one extra read is the hidden User Information List, which SPSync uses to put names to person columns.

Can an AI tool get read-only access to one site?

The request often arrives as "Sites.Selected for the AI tool, read only". Whether that is possible depends on the tool. Some connectors cannot use a Selected scope at all: OpenAI's help article "SharePoint app and setup in ChatGPT" has an Entra administrator approve Sites.Read.All and Files.Read.All, and says Sites.Selected is not supported for live SharePoint search.

With SPSync in the middle, the AI tool needs no Graph permission. SPSync holds a read grant on the named site or list and writes the rows to a database you own. The assistant or agent gets a database login with SELECT on the tables or views it needs, and reads through a third-party database MCP server, SPSync's own read-only spsync mcp serve or its own SQL tool. A third-party server is a separate product you choose. spsync mcp serve reads the list definitions from SharePoint once when it starts, using SPSync's own sign-in on the machine where it runs, then answers from the replica only; the assistant never holds that credential. The one caveat to put in the review is the one above: SharePoint item permissions are not carried into the replica, so the database login is the boundary.

Bring your own app registration

Some organisations require every integration to run under an app registration they own. SPSync supports that. Register an app in your tenant with the same permissions described above, authenticate it with a certificate, and sign in with spsync auth login --client-id <your-client-id> --tenant <your-tenant-id>. SPSync's own credential is always a certificate, never a shared secret in a config file, and we recommend the same for yours.

Is SPSync affected by the SharePoint app-only (ACS) retirement?

No. Older SharePoint tools signed in as an app principal issued by Azure Access Control Services (ACS), the model behind SharePoint Add-ins and many app-only scripts. Microsoft's retirement announcement says ACS stopped working for new tenants on 1 November 2024 and was fully retired for existing tenants on 2 April 2026, with modern authentication through Microsoft Entra ID as the replacement. SPSync has only ever used an Entra app registration and Microsoft Graph, so nothing in this pack depends on ACS. If an earlier export script or connector in your tenant stopped working in April 2026, check whether it signed in through ACS.

A pack filled in for your tenant

This page is the general version. On the Connect screen, the Admin permissions pack card

produces a copy as Markdown or PDF with your tenant name, the app's client id, the exact sites and lists picked, and the grant commands with those ids already in place. Once the grants are made, spsync doctor checks the token and each site grant. The admin permissions docs cover the same ground in more detail. For an AI agent your developers built outside Copilot, see governing a custom agent on SharePoint data.

Full, up to date version: https://spsync.io/resources/admin-permissions-pack/