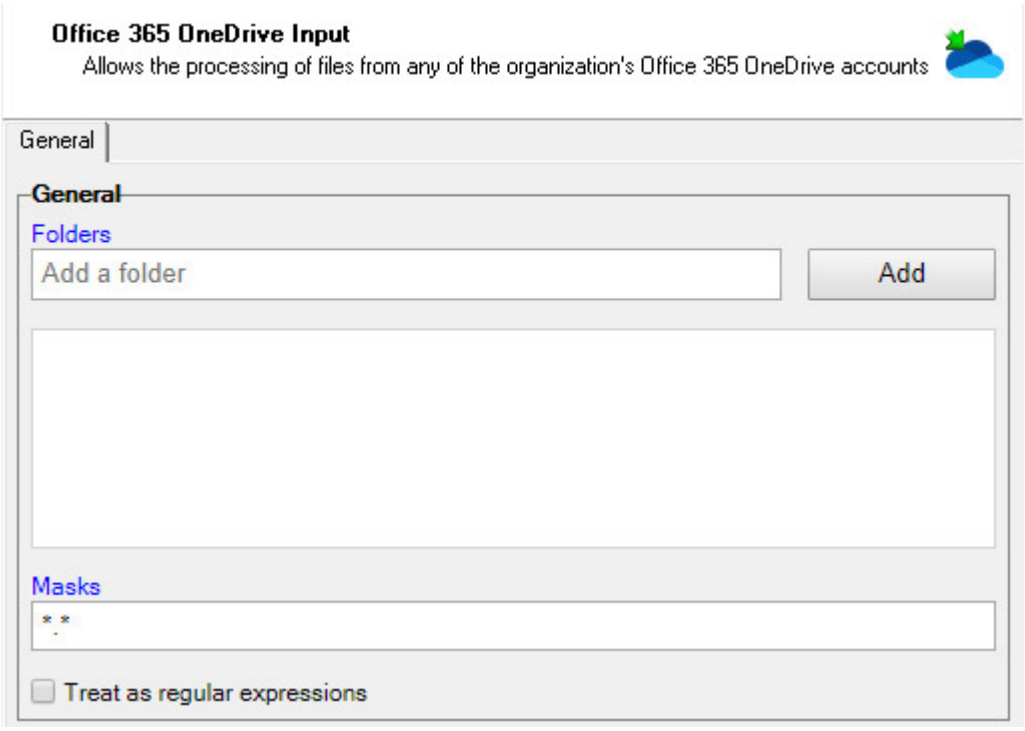
Microsoft 365 Input and Output

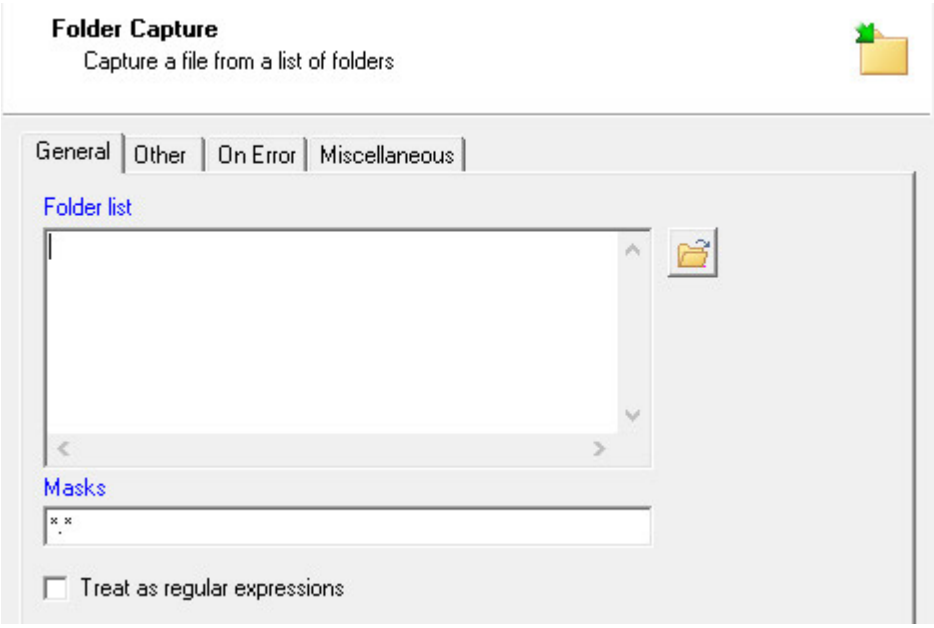
Cloud computing is here to stay and OL Connect Workflow is adapting to this reality. Given the popularity of online storage and mail services, it's only natural that Workflow would have the ability to fetch files from and send them to such services. With OL Connect 2020.1, we have added connectivity to Microsoft 365's OneDrive and Email services. However, this requires your IT department to provide permissions to Workflow. This article explains how to proceed.

The new Workflow tasks

Four Microsoft 365 tasks are now available once you install OL Connect 2020.1: OneDrive Input/Output and Email Input/Output. Those tasks have been modeled after their "on-premise" equivalents and they work roughly the same way, so we're not going to spend much time on the actual feature set of these tasks. For instance, the OneDrive Input task allows you to specify one or more folders to monitor along with file masks, just like the standard Folder Capture task does.

Here's a couple of screenshots demonstrating how similar the old and new tasks are:





In a similar fashion, the Microsoft 365 Email input task works almost exactly the same as the Secure Email input task.

So if you are familiar with the Workflow's standard tasks, you should be able to figure out the new Microsoft 365 tasks in no time.

But... you will first have to involve your IT department.

Why IT is required

Let's imagine that an organization wants to produce a monthly personalized document that is automatically stored in each of its users' OneDrive account. In order to do that, Workflow must be granted access to all users' OneDrive accounts, otherwise it won't be able to store the document in each individual user's OneDrive.

If your IT department were to create a OneDrive account for Workflow, then Workflow would only be able to store files in that specific account, not in each of the OneDrive accounts of the organization. Obviously, IT could request that all users share one of their own OneDrive folders with the Workflow account, which would then allow Workflow to drop files in that specific folder (this is called a *delegated permission*). But that would require each user to be diligent and make sure they share the proper folder, with the proper access rights. For large organizations, this could quickly become an IT nightmare filled with support calls from users who never received their monthly document.

Fortunately, Microsoft 365 provides a higher level of access to an organization's accounts through *application permissions*. These types of permissions are necessary for applications such as backup or anti-virus software that run at the organization level and need access to all the data in all accounts.

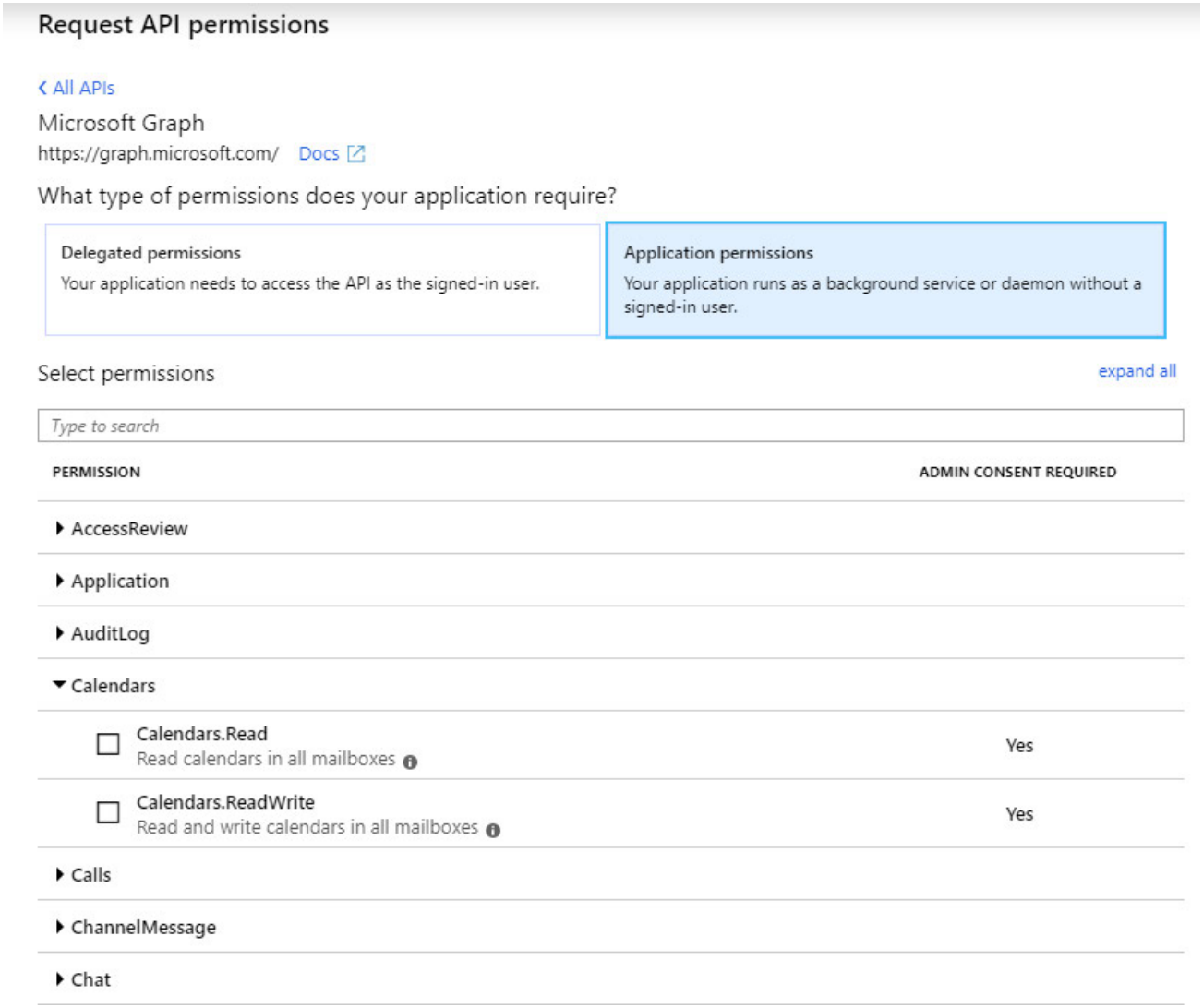
However, an organization's IT department won't grant these types of permissions unless there's a *really* good reason for it: after all, the application could gain access to highly sensitive data across all accounts in the

organization. Still, if the business process warrants it, then IT will look at the list of permissions it can grant and will hand-pick the minimal set that will allow an application like Workflow to perform its intended tasks, but without necessarily granting it full access to everything.

What does IT need to do

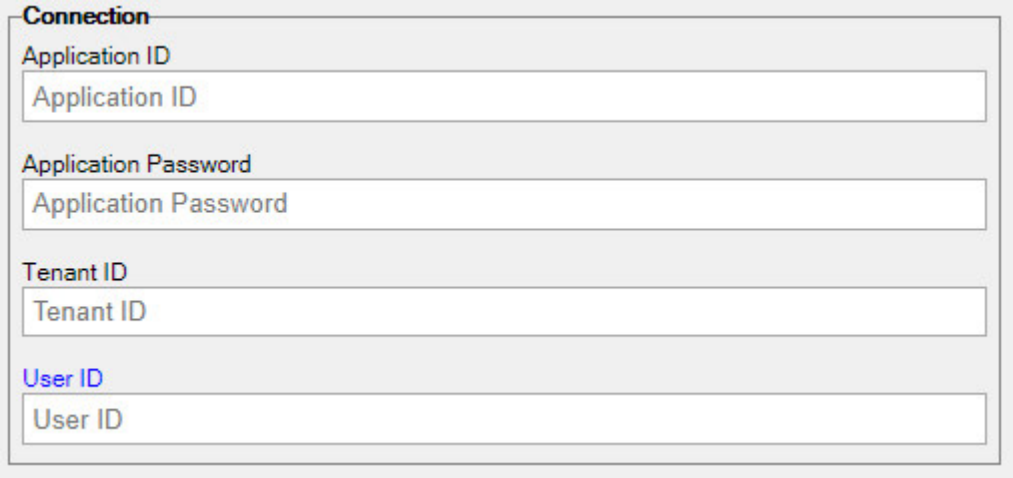
Well first, you have to tell them that Workflow's Microsoft 365 tasks all use the **Microsoft Graph API**. This will immediately tell them where and how they need to grant permissions in the organization's Azure instance.

For instance, the Azure permissions page might look something like this:



Microsoft provides detailed information about permissions on this web page (the picture above was shamelessly lifted from that page!).

At a minimum, IT will need to provide *read access* to the **Users** category (**User.Read.All**) so that the Workflow tasks can at the very least identify the users in the organization. Then, depending on which Microsoft 365 tasks



The last parameter, which is dynamic, is either the user's ID or email address. This will instruct the task on which account to use.

Going back to our initial example, the Microsoft 365 OneDrive output task could be inserted in a loop, with the User ID changing with each iteration so that each user receives their documents directly in their OneDrive account.

Final notes

Workflow's new Microsoft 365 tasks are extremely powerful. Indeed, they are ***so powerful*** that you *must obtain permissions* from your IT department to use them.

As with any sensitive information, you must not share with anyone the various IDs that IT will provide you for Workflow's Microsoft 365 tasks to run properly. IT may also decide to reset the Application ID or Application Password from time to time to insure against possible information leaks, in which case you will have to modify the parameters in each of your Workflow processes accordingly.