IT managers have many options for setting up and managing Microsoft Office 365. Administrators need to consider how they are going to manage user identities for Office 365 and how they are going to assign licenses to users.

When Microsoft migrated corporate users to Office 365, the Microsoft IT team used a hybrid approach for managing identities. The team also developed an application to automatically assign licenses and verify that users have the licenses that they need to stay productive. With less manual work to do, the team can spend more time adding value to business applications.

Deciding how to manage user and group identities

To save the overhead of running and maintaining their own Microsoft Office productivity applications on premises, many organizations are moving to the cloud and using Office 365. This reduces the cost and complexity of maintaining servers and applications, and makes it easier to quickly and economically provide users with productivity tools. For the same reasons, the Microsoft IT team decided to move all Microsoft corporate users to Office 365.

Microsoft uses a hybrid approach, where some applications continue running on the corporate network while Office applications run in the cloud. A key planning step for the move to Office 365 was deciding how to manage user and group identities in a hybrid environment.

In addition, as many as 2,000 new identities for new employees and vendors are added to the Microsoft corporate directory each week. To ensure that users always have the licenses that they need to do their jobs, the team wanted to be able to quickly assign Office 365 licenses to new users. The team also wanted to quickly detect and correct any errors in the Office 365 license assignments for all users.
Identity management through Azure Active Directory

Office 365 uses a cloud-based user identity and authentication service, Microsoft Azure Active Directory (Azure AD), to manage identities. Microsoft uses on-premises Active Directory Domain Services for identity management. The Microsoft IT team had two options for handling identities for Office 365:

- Manage identities for Office 365 users in Azure AD only. This option wasn’t practical at Microsoft because the on-premises directory in Active Directory would continue to provide access control for applications and assets on the corporate network. Separately managing user identities in two different directories—one on premises and one in the cloud—would create extra work and could introduce errors.
- Manage all user identities in the on-premises directory. In this case, the team would synchronize the on-premises Active Directory store with Azure AD to provide online authentication for Office 365, and would keep the two directories synchronized.

Building an automated licensing solution

The Microsoft IT team selected the second option. User and group identities that are created and stored within the on-premises directory in Active Directory would be kept synchronized with Azure AD. The team also decided to enable single sign-on, so that users enter their password only once when they sign in to the network to access their assets, including Office 365.

To address the challenge of synchronizing directories and maintaining a hybrid directory environment, the Microsoft IT team used Azure Active Directory Connect. This tool performed the initial synchronization, and now keeps the on-premises and cloud directories synchronized as new users are added.

For the initial migration, after the user and group accounts were synchronized between Azure AD and the on-premises directory, Office 365 licenses were assigned to the current users. Given the number of users—more than 250,000—manually licensing them would have been both time-consum ing and prone to human error. To automate this task, the Microsoft IT team used a Windows PowerShell script to assign the appropriate Office 365 licenses to each new user.

The Microsoft IT team then developed a C# application to run as a service and assign licenses to new users as they’re created in Active Directory and then synced to Azure AD. The service also corrects any licensing errors that are introduced manually, such as by a help desk technician or other process. The service performs these tasks at prescribed intervals.

Creating a licensing process at scale

Microsoft currently has an Office 365 Enterprise E3 subscription, which includes Microsoft Office Professional Plus, Skype for Business, Microsoft Office Web Apps, Microsoft SharePoint Online, Microsoft Exchange Online, and other apps and services. To access these apps or services, a user needs to be assigned a specific license. There are three ways to assign user licenses for Office 365 applications and services:

- Manually, either for individual users or groups, by using the Microsoft Online Portal. This option is useful for assigning licenses to a small number of users.
- Automatically, for groups of users, by using the Microsoft Online Services Module for Windows PowerShell. A certain amount of manual intervention is required, which takes time and can introduce errors. Therefore, this option isn’t ideal for regularly recurring licensing tasks for large groups of users.
- Automatically for a large group of users by developing a custom automation application. This is useful for assigning many licenses on a regular schedule as well moving users from one subscription to another when required.

IT Showcase Technical Case Study
To assign licenses manually, a technician selects the users to edit and then selects licenses to assign to them. This operation deletes any previous licenses for the users and replaces them with the selected licenses. For Microsoft IT, this approach to migrating a large number of users to Office 365 wasn’t feasible. To learn about existing tools for manual and bulk licensing, see Office 365: License Users for Office 365 Workloads.

The Microsoft IT team wanted to license users as quickly as possible. To save time and reduce errors, the team wrote a Windows PowerShell script to assign licenses to the initial set of migrated users. Then, to perform ongoing and regularly scheduled licensing operations, the Microsoft IT team developed a C# automation service.

The automation service uses Windows PowerShell scripts to return lists of users who have particular attributes from Azure AD. The service puts these users into security groups. It then assigns licenses to users based on their security group membership. For example, users who are entitled to Exchange Online have a target address that has a specific form on premises that points to their Office 365 mailbox. The script returns these users, and the automation service adds them to a security group for Exchange Online users. The service then assigns an Exchange Online license to any user in this group who doesn’t have one.

To update licenses for Office 365, the automation service that the Microsoft IT team built executes three tasks that run on schedules defined within Windows Task Scheduler.

Task 1. License newly synchronized users for Office 365

Every two hours, Azure AD Connect automatically synchronizes the on-premises directory in Active Directory with Azure AD. Synchronized user accounts include current users who have on-premises Exchange mailboxes and new users who do not. These newly synchronized users receive licenses for all Office 365 workloads except Exchange Online. Exchange Online licenses are assigned as a separate task.

This separation helps prevent an issue that occurs when all users are automatically assigned an Exchange Online license. If a user who has an on-premises Exchange mailbox in another forest or Exchange environment is assigned an Exchange Online license, and if the ExchangeGUID value is not present on the Office365 object, a new mailbox is created in Office 365. If this happens, the user can no longer access his or her on-premises mailbox. Therefore, the on-premises mailbox is manually migrated to Exchange Online, and then the Microsoft IT team licenses the user for Exchange Online through a separate task.

Task 2. License new Office 365 users for Exchange Online

The automation service performs this task once daily. It captures users who have specific target addresses and adds them to an Azure AD dynamic security group that contains all users who are entitled to Exchange Online. The members of this security group are then issued the licenses that they need, including the license for Exchange Online.

Task 3. Verify that users have a complete set of licenses, and assign licenses as needed

The automation service audits all corporate users for appropriate Office 365 licenses three times per week. It does this by checking an Azure AD dynamic security group that captures all company employees. An example of such a security group is one that contains accounts that have the IsLicensed flag set to True. This audit corrects any licensing errors. Licensing errors can occur when, for example, a technician assigns licenses during Exchange mailbox migration. If, because of human error, the technician does not assign a complete set of licenses, the automation service assigns the licenses.

Implementing an automation service

The licensing automation service connects to Azure AD and gets a list of users. The service then checks the list of licenses that are currently enabled for each user in Azure AD and assigns licenses to them as needed.

IT Showcase Technical Case Study
The technical components used for the automation service solution are as follows:

- **Graph API.** Interfaces with the back end of Azure AD and Office 365.
- **Azure Active Directory PowerShell Module.** Interacts remotely with portions of the Azure AD back end to retrieve account properties.
- **Automation Service Account.** Has the User Management Administrator role in Office 365 and executes all licensing commands on behalf of the automation service application.
- **Windows Task Scheduler.** A built-in Microsoft Windows component that enables task scheduling within Windows. It starts different instances of the license automation service at various intervals.
- **Windows Server.** Hosts the automation service application.
- **Office 365 subscriptions.** Contain the collection of licenses that should be assigned to all employees at Microsoft.

The initial steps that the Microsoft IT team took to create the automation service were as follows:

- Determine the basic set of Office 365 licenses that every user should have.
- Determine how to retrieve stock keeping unit (SKU) ID and service plan ID information from Azure AD for the Office 365 tenant (in this case, Microsoft). For details about getting these SKUs by using Windows PowerShell cmdlets, see [Get-MsolAccountSku](#).
- Create a User Account Administrator account that has permissions to search on Azure AD, and to get basic user and group information. For instructions, see [Assigning admin roles in Office 365](#).
- Create an application object in Azure AD that has read/write permissions on the directory. The application gets user information and sets licenses. For more information about doing this, see [Integrating Applications with Azure Active Directory](#) and [Building Web Applications for Azure AD](#).
- Create a config.xml file that contains information that will run in a Windows PowerShell script that retrieves the list of users to license. See [Automated Office365 Licensing](#). The automation service reads the config.xml file. The following table shows the relevant variables and values in config.xml.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>tenantname (single value)</td>
<td>The name of the Office 365 tenant.</td>
</tr>
<tr>
<td>tenantid (single value)</td>
<td>The name or globally unique identifier (GUID) representation of the tenant or the tenant GUID.</td>
</tr>
<tr>
<td>clientid (single value)</td>
<td>AppPrincipalID value for the application that has read/write permissions on the directory in Azure AD.</td>
</tr>
<tr>
<td>clientsecret (single value)</td>
<td>The client secret passcode that Azure AD produces when the application is created in Azure AD.</td>
</tr>
<tr>
<td>adminupn (single value)</td>
<td>The UserPrincipalName value of the User Account Administrator account.</td>
</tr>
<tr>
<td>adminpassword (single value)</td>
<td>The password for the User Account Administrator account.</td>
</tr>
<tr>
<td>groupbased (single value)</td>
<td>A True/False value that determines whether to license all unlicensed users (False) or only users in a specified security group (True).</td>
</tr>
<tr>
<td>aadgroupobjectid (single value)</td>
<td>The security group object ID from Azure AD for the security group to license. This variable applies only if groupbased is set to True.</td>
</tr>
<tr>
<td>skuid (single value)</td>
<td>The SKU ID for the license SKU to apply to the selected users.</td>
</tr>
</tbody>
</table>

This variable applies only if groupbased is set to True.
<table>
<thead>
<tr>
<th>Variable</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>enabledplanid (multiple values)</td>
<td>The service plan ID (GUID) for the service plan to enable for selected users. You can specify multiple plans.</td>
</tr>
<tr>
<td>removeskuid (multiple values)</td>
<td>The SKU ID for any licenses to remove from these users when the new SKUs are enabled. Use this variable when you are moving users from one SKU to another and the SKUs have conflicting plans.</td>
</tr>
</tbody>
</table>

- Configure a computer to run the automation service. The Microsoft IT automation service runs on an x64-based computer that has the following configuration.

**Table 2. Automation service computer configuration**

<table>
<thead>
<tr>
<th>Host server</th>
<th>Drives</th>
<th>Software</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Microsoft IT runs two virtual machines) X64 Intel Xeon CPU Operating system: Microsoft Windows Server 2012 R2 32 gigabytes (GB) of RAM</td>
<td>Operating system (primary): 100 GB User storage (secondary): 50 GB Application (tertiary): 300 GB</td>
<td>Microsoft Online Services Sign-in Assistant Azure Active Directory Module for Windows PowerShell</td>
</tr>
</tbody>
</table>

The Microsoft IT team then developed the automation service to function as follows.

**Step 1. Get an authentication token**
By using an application object that has read and write permissions to Azure AD, the automation service gets a token from Azure AD that enables it to authenticate against Azure AD. The token contains the entitlements that are required for every call to Azure AD to get information or assign licenses.

**Step 2. Assign usage location**
The automation service uses the authentication token to make a call to Azure AD, get the usage location for the tenant from Azure AD, and assign it to the user. A tenant can have only one location. When a new user is created in Azure AD, the location field is blank, and a location must be assigned before Office 365 licenses can be assigned.

**Step 3. Get users**
Under User Account Administrator credentials, the automation service connects to the Windows PowerShell web service for the tenant and gets the users that it needs to evaluate for licensing. It uses the Graph application programming interface (API) to return a list of users according to the variables in the config.xml file. If groupbased is set to False, the service reads the read-only Azure AD IsLicensed attribute and returns a list of all users for whom this attribute is False. This is the list of users who have no licenses assigned. If groupbased is set to True, the service returns a list of recently migrated users or the full company list, depending on the value that is set for aadgroupobjectid.

**Step 4. Determine whether each returned user needs a license**
For each user returned in step 3, the automation service searches by user principal name (UPN) and gets a list of the user’s current licenses. The automation service then checks the licenses against the list of licenses in config.xml (the licenses that the user should have).

**Step 5. Assign licenses**
If the two lists in step 4 don’t match, it means that the user doesn’t have the right set of licenses. In this case, the automation service assigns all of the licenses in the config.xml file list to the user by using the Graph API for Azure AD. This overwrites the list of licenses that the user currently has, and ensures that the set of licenses is correct and complete. The automation service iterates through users in the list in this manner, first getting the list of licenses that are currently enabled for the user and then assigning licenses to the user as necessary. For Graph API documentation for Azure AD, see Quickstart for the Azure AD Graph API.

To download sample code that the Microsoft IT team developed to automate Office 365 licensing at Microsoft, see Automated Office365 Licensing.

Converting users from Enterprise E3 to Enterprise E5

In the near future, Microsoft will be moving from the Enterprise E3 subscription to the Enterprise E5 subscription. To do this, we’ll have to analyze our current user setup to verify which users have which licenses. In our automation, we will then replace the skuid attribute in the config.xml files with the new SKU ID number for E5. We’ll also replace the current enabledplanids attribute with the service plan IDs of the corresponding services within the E5 SKU—Exchange for Exchange and SharePoint for Microsoft SharePoint. We will then add the E3 SKU ID number to the removeskuid attribute list. This will ensure that, the next time the automation runs, it will remove users from the E3 subscription and enable them on E5.

Note that this process doesn’t roll out any new services within E5, but only moves the current services over from the E3 subscription to the E5 subscription. Rolling out new E5 features will require further analysis to determine whether we can add the features to all users, or whether we need to add them to a subset of users, as we did for the Exchange Online licenses.

Logging

To support investigation, forensics, security auditing, and tech support, the Microsoft IT team designed the automation application to log every write to Azure AD and log every error. The team uses use logs to track performance, troubleshoot issues, and research issues when users contact the help desk for assistance. A user can encounter issues if, for example, a technician made an error while editing licenses, or if the user hasn’t been licensed at all.

Azure AD also provides audit logs, in the form of reports that are available from the Office 365 admin center. The following table lists these reports.

<table>
<thead>
<tr>
<th>Table 3. Reports available from the Office 365 admin center</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Report</strong></td>
</tr>
<tr>
<td>Sign ins from unknown sources</td>
</tr>
<tr>
<td>Sign ins after multiple failures</td>
</tr>
<tr>
<td>Sign ins from multiple geographies</td>
</tr>
</tbody>
</table>
Report | Use it to do this
---|---
Account provisioning errors | Monitor errors that occur during the synchronization of accounts from software as a service (SaaS) applications to Azure AD. Entries in this report might indicate an issue with a user’s ability to access external applications.
Audit | View the Azure AD audit log. This report contains entries for events such as creating a new user account, changing the properties of a user account, or changing a user password. Each entry includes the date and time of the event, the user who made the change, the change that was made, and the user account that was changed. Entries in this report are kept for 30 days.

The Microsoft IT team uses the Audit report extensively to get a snapshot view of user activity and licensing. Sometimes, a user complains that one of his or her applications isn’t available, or the automation service must license a user repeatedly because the user is repeatedly being de-licensed. When this happens, the team looks in the Audit report to see whether the licenses are being manipulated by, for example, another user, a service account, or another Azure AD application that was given rights to make edits.

For details about the available audit log reports, see Use Azure Active Directory sign-in and audit reports and View your access and usage reports.

Benefits
The Microsoft IT team met its goals for the migration to Office 365. The migration was largely transparent to users, who were able to continue using their familiar Office tools without interruption. New users have been able to become productive more rapidly than before the migration because automation has made the licensing process faster and less prone to errors. When errors do occur, most of them are quickly and automatically corrected. Fewer user issues end up in a help desk call and, thanks to logging, it’s easier for Microsoft IT staff to resolve user issues by researching the log files. Therefore, the Microsoft IT team is freed up to do more value-added work that benefits the business.

Best practices
Here are some of the best practices that the Microsoft IT team learned while synchronizing user and group accounts and licensing Office 365 users.

Resolve duplicate attributes in your on-premises Active Directory
Before the Microsoft IT team could synchronize the on-premises and cloud directories, it needed to resolve duplicate attributes in the on-premises directory. The directory had been used for a long time and had accumulated many duplicate user and group attributes. Although this wouldn’t create an issue in Active Directory, the attributes had to be cleaned up to ensure successful synchronization with Azure AD.

In an on-premises environment, duplicate aliases are acceptable, provided that they have unique suffixes after the at sign (@). However, all aliases in Office 365 must be unique for a given organization. Even if addresses have multiple unique suffixes, all aliases must be unique in the Simple Mail Transfer Protocol (SMTP) address. Before migration, the Microsoft IT team needed to resolve about 30,000 duplicates in the on-premises directory. This project took six weeks, but the migration would have failed without it.

As a best practice, before you migrate to Azure AD, make sure that alias values are unique within your on-premises Active Directory. Azure AD will not accept duplicate values. Any synchronization operation that tries to write a duplicate alias value to the directory will fail. For details about resolving duplicate attributes, see Duplicate or invalid attributes prevent directory synchronization in Office 365.
Log all writes and errors
To support forensics, security auditing, and tech support, it’s a good idea to log every write that the automation application makes to Azure AD and log every error. Logging helps you track performance, troubleshoot problems, and research issues when users contact the help desk for assistance.

Account for throttling in your code
Azure AD uses throttling to ensure that no single tenant can cause issues with the entire service. To handle this potential for throttling, include code in your automation application that looks for a throttling error and pauses automation for five minutes before restarting.

Search on unique attributes
When you search Azure AD for users, search on the UPN or some other unique attribute. For example, if the display name is used instead of the UPN, results could include multiple users who have the same name, whereas using the UPN or some other unique attribute will return only a single user.

Resources
Understanding Office 365 identity and Azure Active Directory
Microsoft Office 365 business subscription plans
Use Azure Active Directory sign-in and audit reports
Optimizing network performance for Microsoft Office 365