

Azure Active Directory **User Provisioning** Deployment Plan

**How to use this guide**

This step-by-step guide walks through the implementation of User Provisioning in a five-step process. The links below take you to each of those steps.



**1**

**[Include](#_Stakeholders_and_Sign-off)**

[Stakeholders](#_Stakeholders_and_Sign-off)

**2**

**[Plan](#_Planning_Your_Implementation)**

[Your project](#_Planning_Your_Implementation)

**3**

**[Design](#_Design)**

[Policies and integration](#_Design)

**5**

**[Manage](#_Operationalize_your_Implementation)**

[Your implementation](#_Operationalize_your_Implementation)



**4**

**[Implement](#_Implementing_Your_Solution_1)**

[Your design](#_Implementing_Your_Solution_1)

**Note:**

Throughout this document, you will see items marked as

* **Microsoft Recommends**

These are general recommendations, and you should only implement if they apply to your specific enterprise needs.

Table of Contents

Contents

[Introduction 1](#_Toc509497549)

[Purpose of Document 1](#_Toc509497550)

[Confidentiality Statement 1](#_Toc509497551)

[Azure Active Directory User Provisioning 1](#_Toc509497552)

[Current State 1](#_Toc509497553)

[Current Authentication Methods 1](#_Toc509497554)

[Costs of Current Authentication 1](#_Toc509497555)

[Benefits of Azure Active Directory automatic user provisioning for <<Application Name>> 2](#_Toc509497556)

[Comparing Azure Active Directory automatic user provisioning to other solutions 2](#_Toc509497557)

[How to use this guide 1](#_Toc509497558)

[Stakeholders and Sign-off 2](#_Toc509497559)

[Planning Your Implementation 3](#_Toc509497560)

[In Scope 3](#_Toc509497561)

[Out of scope 3](#_Toc509497562)

[Tracking Timelines 4](#_Toc509497563)

[Licensing 4](#_Toc509497564)

[Solution Architecture Diagrams and Description 6](#_Toc509497565)

[Planning for Automatic User Provisioning 9](#_Toc509497566)

[Designing Your Implementation 12](#_Toc509497567)

[Scoping requirements 12](#_Toc509497568)

[Implementing Your Solution 15](#_Toc509497569)

[Phase 1: Configuring automatic user provisioning 15](#_Toc509497570)

[Phase 2: User Acceptance Testing (UAT) 16](#_Toc509497571)

[Phase 3: Transitioning into production 16](#_Toc509497572)

[Phase 4: Rollback steps 16](#_Toc509497573)

[Operationalize your Implementation 17](#_Toc509497574)

[Reporting and monitoring 17](#_Toc509497575)

[Troubleshooting 18](#_Toc509497576)

[Reference Documentation 19](#_Toc509497577)

# Business Value of User Provisioning

This document presents an executive summary of the business case for moving forward with enabling Azure Active Directory automatic user provisioning for «APPLICATIONNAME» (“The Application”).

Many organizations rely upon software as a service (SaaS) applications such as Office 365, Box and Salesforce for end user productivity. Historically, IT staff have relied on manual provisioning methods or custom scripts to securely manage user identities in each SaaS application.

Azure Active Directory User Provisioning simplifies this process by securely automating the creation, maintenance, and removal of user identities in cloud (SaaS) applications based on business rules. This allows an enterprise to effectively scale their identity management systems on both cloud-only and hybrid environments as they expand their dependency on cloud-based solutions.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | | \\MAGNUM\Projects\Microsoft\Cloud Power FY12\Design\ICONS_PNG\Increase.png | **INCREASE PRODUCTIVITY**  Simplify the management of user identities across SaaS applications by providing a single user provisioning management interface. This includes having a single set of policies to determine who is provisioned, who can sign into an application and what user information to provision. | | \\MAGNUM\Projects\Microsoft\Cloud Power FY12\Design\ICONS_PNG\Confidentiality.png | **MANAGE RISK**  Secure your organization by ensuring that user identities and access to key SaaS apps are automatically updated when users transition or leave the organization. This can be implemented based on a user’s employee status or groups that define user roles and/or access. | | C:\Users\mitchellg\Desktop\Simple_Licensing.png | **ADDRESS COMPLIANCE AND GOVERNANCE**  Supports native audit logs for every user provisioning request performed by each application for both source and target systems. This includes user imports, exports, and synchronization. | | \\MAGNUM\Projects\Microsoft\Cloud Power FY12\Design\ICONS_PNG\Within_Your_Reach.png | **MANAGE COST**  Reduce costs by avoiding inefficiencies and human error associated with manual provisioning. This includes maintaining custom-developed user provisioning solutions, scripts, and audit logs. | |  |

**Comparing Azure Active Directory automatic user provisioning to other solutions**

|  |  |
| --- | --- |
| User provisioning options | Advantages of automatic user provisioning |
| Manual provisioning | * Automates join, move, leave scenarios * Reduces human labor and error, improves security |
| SAML Just-in-Time provisioning | * Adds support for leaver scenarios, groups * Reduces human labor and error for deprovisioning, improves security |
| Custom solutions | * Hosted in the cloud, reduces infrastructure costs * Pre-build connectors, no code required |
| Other IDaaS solutions | * Doesn’t require deployment of another cloud directory * Leverages what we already get with Office 365 |

## Stakeholders and Sign-off

The following section serves to identify all the stakeholders that are involved in the project and need to sign off, review, or stay informed. Add stakeholders to the table below as appropriate for your organization.

* + SO = Sign-off on this project
  + R = Review this project and provide input
  + I = Informed of this project

|  |  |  |
| --- | --- | --- |
| Name | Role | Action |
| Enter name and email | **IT Support Manager**  *A representative from the IT support organization who can provide input on the supportability of this change from a helpdesk perspective.* | SO |
| Enter name and email | **Identity Architect or Azure Global Administrator**  *A representative from the identity management team in charge of defining how this change is aligned with the core identity management infrastructure in the customer’s organization.* | SO |
| Enter name and email | **Application Business Owner**  *A representative colleague who can provide input on the user experience and usefulness of this change from an end-user’s perspective and owns the overall business aspect of the application, which may include managing access.* | SO/I |
| Enter name and email | **Security Owner**  *A representative from the security team that can sign off that the plan will meet the security requirements of your organization.* | SO |

# Planning Your Implementation

Proper planning is essential to any successful integration in an IT environment. This section will guide you through planning steps that will help simplify your organization’s onboarding of automatic user provisioning for <<APPLICATION NAME>>.

### In Scope

The following is in scope for this project:

* Enabling end-to-end outbound user provisioning to <<APPLICATION NAME>>, so that a user created and/or updated in our organizational Active Directory infrastructure is also automatically synchronized to the application, in accordance with business policies.
* Enable leveraging of existing user attributes from corporate Active Directory in Azure Active Directory for new accounts within the application.
* Enabling the support organization to support and manage this new change, ensuring the right helpdesk processes are in place to ensure on-going end-user success.
* Documenting and testing a recovery plan.
* Approving a business continuity plan.
* Approving an information security risk assessment.
* Designing operational support for the production service.
* The following environments are in scope for this design:
  + Production
  + Test / QA

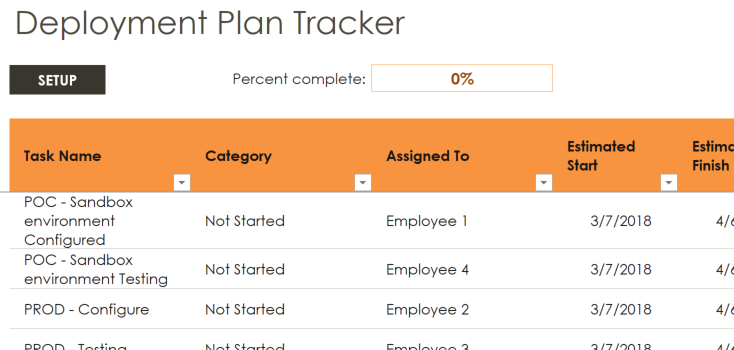
### Out of scope

The following are out of scope of this project:

* Extending the corporate Active Directory system with any additional or new attributes that are require by the application. Any new attributes necessary will be created in the Azure Active Directory.
* Disabling of the existing federation relationship between the application and our corporate federation solution.
* Enabling single sign-on to the application using Microsoft Azure Active Directory federation technologies.
* Extension of on-premises AD to include new attributes which will be provisioned to the Azure AD or application environments.
* Enabling inbound user provisioning from applications to the Active Directory infrastructure.

### Tracking Timelines

Tracking your plan is an important aspect of project success. You may use the embedded Deployment Plan Tracker spreadsheet below to monitor and schedule your committed timelines for this project. Begin tracking additional items as you progress through the deployment plan that may require an action or prerequisite:



### Licensing

#### Azure Active Directory Licensing

You will need an [Azure AD License.](https://azure.microsoft.com/pricing/details/active-directory/) The number of objects in your directory and the features you wish to deploy will affect your licensing choices. While many features are included with Azure Free and Azure Basic, some features require Azure AD Premium (P1 or P2). Common Azure AD scenarios include the following recommended security features:

* [User Provisioning](https://azure.microsoft.com/pricing/details/active-directory/) (Free)
* [Group Provisioning](https://azure.microsoft.com/pricing/details/active-directory/) (Basic Required)
* [Conditional Access (CA)](https://docs.microsoft.com/azure/active-directory/active-directory-conditional-access-azure-portal) (P1 Required)
* [Azure Multi-Factor Authentication (MFA) (P1 Required)](https://docs.microsoft.com/azure/multi-factor-authentication/multi-factor-authentication-how-it-works)
* [Group based membership](https://docs.microsoft.com/azure/active-directory/active-directory-manage-groups) (P1 required)
* [Identity Protection](https://docs.microsoft.com/azure/active-directory/active-directory-identityprotection) (P2 Required)

The following table describes some of the license requirements that may be relevant. For a full list of license requirements, click [here](https://azure.microsoft.com/pricing/details/active-directory/).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LICENSE TYPE | FREE | BASIC | PREMIUM P1 | PREMIUM P2 |
| Directory Objects | 500,000 Object Limit | No Object Limit | No Object Limit | |
| Single Sign-On | 10 apps per user (pre-integrated SaaS and developer-integrated apps) | 10 apps per user (free tier + Application proxy apps) | No Limit (free, Basic tiers + Self-Service App Integration templates) | |
| Group Provisioning | Not Available | Available | Available | |
| User Provisioning | Available | | Available | |
| Bring your own application SCIM | Available | | Available | |
| Access based on group membership | Not Available | | Available | |
| CA based on group and location | Not available | | Available | |
| CA based on device state (Allow access from managed devices) | Not available | | Available | |

If you have an existing Enterprise Mobility and Security (EMS) subscription with Microsoft, you may already have Azure AD Premium.

**Enterprise Mobility and Security (EMS) subscriptions:**

* EMS E3 includes P1
* EMS E5 includes P2.

If you have an existing Enterprise Agreement or Server and Cloud Enrollment, you may already have Azure Premium. Check the details of your agreement.

#### Application Licensing

You will also need the appropriate license for your application to meet your business needs.

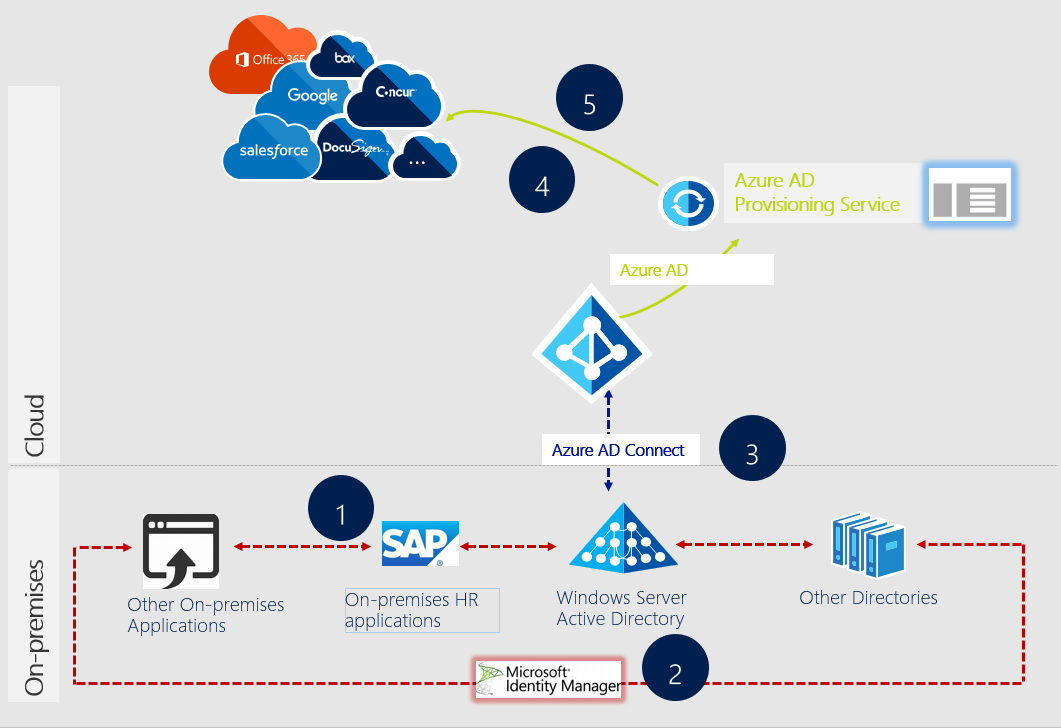
Discuss with the application owner whether the users assigned to and accessing the application have the appropriate licenses for their roles within the application. If Azure AD manages the automatic provisioning based on roles, the roles that are assigned in Azure AD must align with the correct number of licenses owned within the application. Improper number of licenses owned in the application may lead to errors during the provisioning/updating of a user.

## Solution Architecture Diagrams and Description

The topologies for outbound automatic user provisioning are represented below:

#### Azure AD Outbound Automatic User Provisioning – Hybrid Enterprises

The following diagram illustrates the end-to-end user provisioning workflow that occurs for common hybrid environments. In this example, user creation occurs in an HR database connected to an on-premises directory while outbound automatic user provisioning is managed by the Azure AD provisioning service to the target SaaS applications:

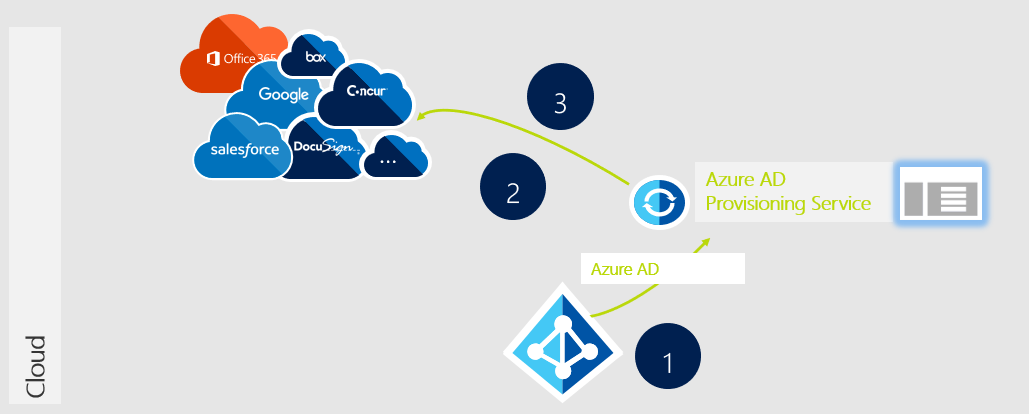


**Description of workflow:**

1. Users/groups are created in an on-premises HR application/system, such as SAP
2. A service (e.g. MIM) runs scheduled synchronizations of identities from the HR system to a local Active Directory and any other directories or local applications.
3. Azure AD Connect synchronization service runs scheduled synchronizations of identities (users & groups) from the local Active Directory to Azure Active Directory.
4. Azure AD provisioning service begins an initial sync which will:
   1. Query all users and groups from the source system, retrieving all attributes defined in the [attribute mappings](https://docs.microsoft.com/azure/active-directory/active-directory-saas-customizing-attribute-mappings).
   2. Filter the users and groups returned, using any configured [assignments](https://docs.microsoft.com/azure/active-directory/active-directory-coreapps-assign-user-azure-portal) or [attribute-based scoping filters](https://docs.microsoft.com/azure/active-directory/active-directory-saas-scoping-filters).
   3. When a user is found to be assigned or in scope for provisioning, the service queries the target system for a matching user using the designated [matching attributes](https://docs.microsoft.com/azure/active-directory/active-directory-saas-customizing-attribute-mappings#understanding-attribute-mapping-properties).
   4. If a matching user is not found in the target system, it is created using the attributes returned from the source system.
   5. If a matching user is found, it is updated using the attributes provided by the source system.
   6. If the attribute mappings contain "reference" attributes, the service performs additional updates on the target system to create and link the referenced objects.
   7. Persist a watermark at the end of the initial sync, which provides the starting point for the subsequent incremental syncs.
5. Upon a successful completion of an initial sync, the Azure AD provisioning service will:
   1. Query the source system for any users and groups that were updated since the last watermark was stored.
   2. Filter the users and groups returned, using any configured assignments or attribute-based scoping filters.
   3. When a user is found to be assigned or in scope for provisioning, the service queries the target system for a matching user using the designated matching attributes.
   4. If a matching user is not found in the target system, it is created using the attributes returned from the source system.
   5. If a matching user is found, it is updated using the attributes provided by the source system.
   6. If the attribute mappings contain "reference" attributes, the service performs additional updates on the target system to create and link the referenced objects.
   7. If a user that was previously in scope for provisioning is removed from scope (including being unassigned), the service disables the user in the target system via an update.
   8. If a user that was previously in scope for provisioning is disabled or soft-deleted in the source system, the service disables the user in the target system via an update.
   9. If a user that was previously in scope for provisioning is hard-deleted in the source system, the service deletes the user in the target system. In Azure AD, users are hard-deleted 30 days after they are soft-deleted.
   10. Persist a new watermark at the end of the incremental sync, which provides the starting point for the subsequent incremental syncs.

#### Azure AD Outbound Automatic User Provisioning – Cloud-only Enterprises

The following diagram illustrates the end-to-end user provisioning workflow that occurs for common cloud-only environments. In this example, user creation occurs in Azure AD and the automatic user provisioning is managed by the Azure AD provisioning service to the target (SaaS) applications:



Description of workflow:

1. Users/groups are created in Azure AD.
2. Azure AD provisioning service begins an initial sync which will:
   1. Query all users and groups from the source system, retrieving all attributes defined in the [attribute mappings](https://docs.microsoft.com/azure/active-directory/active-directory-saas-customizing-attribute-mappings).
   2. Filter the users and groups returned, using any configured [assignments](https://docs.microsoft.com/azure/active-directory/active-directory-coreapps-assign-user-azure-portal) or [attribute-based scoping filters](https://docs.microsoft.com/azure/active-directory/active-directory-saas-scoping-filters).
   3. When a user is found to be assigned or in scope for provisioning, the service queries the target system for a matching user using the designated [matching attributes](https://docs.microsoft.com/azure/active-directory/active-directory-saas-customizing-attribute-mappings#understanding-attribute-mapping-properties).
   4. If a matching user is not found in the target system, it is created using the attributes returned from the source system.
   5. If a matching user is found, it is updated using the attributes provided by the source system.
   6. If the attribute mappings contain "reference" attributes, the service performs additional updates on the target system to create and link the referenced objects.
   7. Persist a watermark at the end of the initial sync, which provides the starting point for the subsequent incremental syncs.
3. Upon a successful completion of an initial sync, the Azure AD provisioning service will:
   1. Query the source system for any users and groups that were updated since the last watermark was stored.
   2. Filter the users and groups returned, using any configured assignments or attribute-based scoping filters.
   3. When a user is found to be assigned or in scope for provisioning, the service queries the target system for a matching user using the designated matching attributes.
   4. If a matching user is not found in the target system, it is created using the attributes returned from the source system.
   5. If a matching user is found, it is updated using the attributes provided by the source system.
   6. If the attribute mappings contain "reference" attributes, the service performs additional updates on the target system to create and link the referenced objects.
   7. If a user that was previously in scope for provisioning is removed from scope (including being unassigned), the service disables the user in the target system via an update.
   8. If a user that was previously in scope for provisioning is disabled or soft-deleted in the source system, the service disables the user in the target system via an update.
   9. If a user that was previously in scope for provisioning is hard-deleted in the source system, the service deletes the user in the target system. In Azure AD, users are hard-deleted 30 days after they are soft-deleted.
   10. Persist a new watermark at the end of the incremental sync, which provides the starting point for the subsequent incremental syncs.

## Planning for Automatic User Provisioning

Azure Active Directory (Azure AD) features pre-integrated user provisioning support for a variety of popular SaaS applications as well as generic user provisioning support for applications that implement specific parts of the System for Cross-Domain Identity Management (SCIM) 2.0 protocol specification.

Applications that support provisioning in the [Azure AD application gallery](https://docs.microsoft.com/azure/active-directory/active-directory-saas-tutorial-list) come pre-configured with default user provisioning settings. However, you have the option to customize the configuration of the user provisioning connector to suit your organization’s needs.

Once configured, Azure AD will be able to send requests to create, modify, deactivate, or delete assigned users and/or groups to the desired applications via their web services. The web services can then translate those requests into operations on the target identity store.

* **Microsoft recommends** utilizing the pre-integrated user provisioning connectors for a SaaS application. If not available, utilize the [BYOA SCIM](https://docs.microsoft.com/azure/active-directory/active-directory-scim-provisioning) generic user provisioning support for SaaS applications.

Below is a list of items that are useful when planning your Azure AD automatic user provisioning implementation:

#### Understand SCIM

SCIM, or System for Cross-domain Identity Management, is an open standard that allows for the automation of user provisioning. SCIM communicates user identity data between identity providers (e.g. Microsoft) and service providers requiring user identity information (e.g. SaaS apps like Salesforce).

* Created in 2011 and experienced growth of adoption in 2015-2016 among popular SaaS apps like Workplace by Facebook, Slack, Cerner, GitHub, etc.
* Majority of the pre-integrated connectors for applications in Azure AD utilize SCIM for user provisioning.
* Azure AD supports a generic SCIM connector (BYOA SCIM) that works with applications implementing a profile of SCIM 2.0 as documented [here](https://azure.microsoft.com/documentation/articles/active-directory-scim-provisioning/).

To learn more about SCIM, refer to <http://www.simplecloud.info/>.

#### Determine <<APPLICATION NAME>’s user provisioning requirements

Even if an application utilizes SCIM, each CRUD (Create, Replace, Update, Delete) operation or attributes/objects used may differ from application to application. Before implementing Azure AD automatic user provisioning, define a list of objects and operations needed based on the list below:

User accounts

* User provisioning operations to be performed on the user objects for the target systems
* Configurable user attribute mappings between source and target systems
* Control how existing users are matched and updated between source and target systems
* Supported user attributes for both source and target systems
* Supported user operations for both source and target systems

Groups (for selected apps)

* Group provisioning operations to be performed on the group objects for the target systems
* Configurable group attribute mappings between source and target systems
* Control how existing groups are matched and updated between source and target systems
* Supported group attributes for both source and target systems
* Supported group operations for both source and target systems

#### Setting up Azure AD automatic user provisioning

Before setting up Azure AD automatic user provisioning, be aware of the following (if applicable) to reduce issues post-deployment:

* Ensure that the admin credentials provided to the Azure AD provisioning service will allow it to connect to the user management API provided by the target system.
* Ensure that the attributes used to map user/group objects between source and target systems are resilient – they should not cause users/groups to be provisioned incorrectly if the attributes change (e.g. user moved to a different part of the company, etc.)
* Some applications may have specific restrictions and/or requirements that need to be met for user provisioning to work correctly (e.g. Slack truncates values for certain attributes, etc.) These are documented in Microsoft’s [automatic user provisioning tutorials](https://docs.microsoft.com/azure/active-directory/active-directory-saas-tutorial-list) specific to each application.
* Confirm schema consistency between source and target systems. Common issues include:
  + Attributes such as **UPN** or **mail** not matching between <<APPLICATION NAME>> and Azure Active Directory due to different formatting (e.g. UPN in Azure AD set as [john\_smith@contoso.com](mailto:john_smith@contoso.com) and in <<APPLICATION NAME>> is [jsmith@contoso.com](mailto:jsmith@contoso.com))

#### Preparing for the initial sync

When the Azure AD provisioning service runs for the first time, it performs an initial sync against the source system and target systems to create a snapshot of all user objects for each target system.

The time taken for initial syncs are directly dependent on how many users, groups and group members are present in the source system. Initial syncs for Azure AD tenants with over 100,000 users and/or group objects combined can take a long time.

* **Microsoft recommends** disabling the sync of group objects in the attribute mappings of your provisioning configuration if you do not want to provision group names and group memberships to your application, which in turn will help speed up the initial sync.

#### Monitoring user provisioning operational health

After a successful initial sync, the Azure AD provisioning service will continue to run back-to-back incremental syncs indefinitely, at intervals defined in the tutorials specific to each application, until one of the following events occur:

* The service is manually stopped, and new initial sync triggered using the Azure portal, or using the appropriate Graph API command.
* A new initial sync is triggered due to a change in attribute mappings or scoping filters.
* The provisioning process goes into quarantine due to a high error rate and stays in quarantine for more than four weeks at which it will automatically be disabled.

To review these events, refer to the provisioning audit logs and reporting which are described [here](https://docs.microsoft.com/azure/active-directory/active-directory-saas-provisioning-reporting).

#### Planning your Security Review

It is common for a security review to be required as part of a deployment of a new service. If a security review is required or has not yet been conducted, please review the many Azure AD [whitepapers](https://www.microsoft.com/download/details.aspx?id=36391) that will provides an overview for the identity as a service.

# Designing Your Implementation

This section is used to assist you in designing the automatic user provisioning implementation in your environment that best meets your business needs. This document will indicate when Microsoft has a recommendation among the choices presented.

## Scoping requirements

Go through the workflow below to scope the key requirements needed to implement automatic user provisioning in your environment:

#### Determine the type of connector to use

Check if <<APPLICATION NAME>> already has a pre-integrated user provisioning connector with Azure AD. You can do so by referring to the[SaaS application integration tutorial list](https://docs.microsoft.com/azure/active-directory/active-directory-saas-tutorial-list) which will list an application tutorial for user provisioning if the target application has a pre-integrated user provisioning connector. Depending on the outcome, your next steps are as below:

|  |  |  |
| --- | --- | --- |
| Scoping Question | Answer | Recommended Next Steps |
| Does <<APPLICATION NAME>> have a pre-integrated user provisioning connector? | Yes | * Move forward to the next step in this workflow to understand [and collect the admin credentials required](#_Collect_the_admin) for automatic user provisioning |
| No | * Create a request [here](https://feedback.azure.com/forums/374982-azure-active-directory-application-requests/category/172035-user-provisioning) for a pre-integrated user provisioning connector. * Work with the application owner to utilize the [BYOA SCIM](https://docs.microsoft.com/azure/active-directory/active-directory-scim-provisioning) generic user provisioning support for SaaS applications. |

#### Collect the admin credentials required

When implementing Azure AD automatic user provisioning, you will need to provide certain admin credentials that are used to connect to the target system’s user management endpoint - to facilitate user provisioning which may differ for each application. Common admin credentials include:

* **Admin Username** - Username for an admin account on the target system.
* **Admin Password** - Password for an admin account on the target system.
* **Secret Token** - An OAuth bearer token from the target system.
* **Tenant URL** - The entire URL of the user management endpoint for the target system.
* **Domain** - The domain or subdomain name of the user management endpoint for the target system.
* **Notification Email** - Email address of a person or a group who should receive user provisioning error notifications.

Use the following tables to document the admin credentials required for <<APPLICATION NAME>>:

|  |  |
| --- | --- |
| Credential Type | Values |
| *e.g. Admin Username* | *e.g. test@contoso.com* |
|  |  |

#### Define required attributes for your environment

To implement automatic user provisioning, you will need to define the user and/or group attributes that are needed by your organization.

**Note:** Unless an application utilizes SCIM, each application may have their own schema for attributes needs.

Use the tables below to document the Azure AD (or AD if applicable) attributes needed along with their expected mappings to the attributes for <<APPLICATION NAME>>. Feel free to extend the tables as needed.

User attributes needed:

|  |  |  |
| --- | --- | --- |
| AD Attribute (if applicable) | Azure AD Attribute | <<APPLICATION NAME>> Attribute |
| *e.g. User Principal Name (UPN)* | *e.g. User Principal Name (UPN)* | *e.g. userName* |
|  |  |  |

Group attributes needed:

|  |  |  |
| --- | --- | --- |
| AD Attribute (if applicable) | Azure AD Attribute | <<APPLICATION NAME>> Attribute |
| *e.g. member* | *e.g. members* | *e.g. memberships* |
|  |  |  |

**Note:**  Azure Active Directory supports attribute mapping by direct attribute to attribute mapping, providing constant values, or [writing expressions for attribute mappings](https://docs.microsoft.com/azure/active-directory/active-directory-saas-writing-expressions-for-attribute-mappings). This flexibility gives you ultimate control to what will be populated in the targeted application attribute.

#### Choose which users and/or groups to synchronize

Before automatic user provisioning can be implemented, you will need to determine the users and/or groups to be synchronized to <<APPLICATION NAME>>. The table below will help you understand and decide which method is best for your needs:

|  |  |  |
| --- | --- | --- |
| Scoping Question | Answer | * Microsoft Recommended Next Steps |
| What is the source system for your automatic user provisioning implementation? | Active Directory | * Utilize [scoping filters](https://docs.microsoft.com/azure/active-directory/active-directory-saas-scoping-filters) as the primary method to determine which users and/or groups are scope in provisioning. * A scoping filter allows the Azure AD provisioning service to include or exclude any users and/or groups who have an attribute that matches a specific value. * Utilize [user and group assignments](https://docs.microsoft.com/azure/active-directory/active-directory-coreapps-assign-user-azure-portal) as needed for additional filtering. |
| Azure Active Directory | * Utilize [user and group assignments](https://docs.microsoft.com/azure/active-directory/active-directory-coreapps-assign-user-azure-portal) as the primary method to determine which users and/or groups are scope in provisioning. * These assignments are also used for enabling single sign-on. * Provides a single method to manage both application access and user provisioning. * Utilize [scoping filters](https://docs.microsoft.com/azure/active-directory/active-directory-saas-scoping-filters) as needed for additional filtering. |

## 

# Implementing Your Solution

This section is used to guide you through the implementation and testing of your automatic user provisioning using your design requirements documented in the previous section. This workflow is divided into four phases.

## Phase 1: Configuring automatic user provisioning

* **Microsoft recommends** The initial configuration of automatic user provisioning should be done on a test environment with a small subset of users before scaling it to all users in production.

If <<APPLICATION NAME>> has a pre-integrated user provisioning connector:

1. Refer to the <<APPLICATION NAME>> specific integration [tutorial](https://docs.microsoft.com/azure/active-directory/active-directory-saas-tutorial-list) to configure its pre-integrated user provisioning connector.
2. Customize your desired user and/or group attribute mappings for <<APPLICATION NAME>> per the [instructions here](https://docs.microsoft.com/azure/active-directory/active-directory-saas-customizing-attribute-mappings).
3. If the data values between your source and target systems are incompatible, you can [configure expressions for attribute mappings](https://docs.microsoft.com/azure/active-directory/active-directory-saas-writing-expressions-for-attribute-mappings) that will convert your users and/or groups data into formats that are more acceptable for <<APPLICATION NAME>>.
4. Configure the desired users and/or groups that you would like to synchronize to <<APPLICATION NAME>> using [user and group assignments](https://docs.microsoft.com/azure/active-directory/active-directory-coreapps-assign-user-azure-portal) and/or [scoping filters](https://docs.microsoft.com/azure/active-directory/active-directory-saas-scoping-filters).

If <<APPLICATION NAME>> does not have a pre-integrated user provisioning connector:

1. Create a new request [here](https://microsoft.sharepoint.com/teams/apponboarding/Apps/SitePages/default.aspx) for a pre-integrated user provisioning connector for <<APPLICATION NAME>>.
2. Work with the application owner to ensure that <<APPLICATION NAME>> is able to utilize the [BYOA SCIM](https://docs.microsoft.com/azure/active-directory/active-directory-scim-provisioning) generic user provisioning support for SaaS applications – this is a requirement for Azure AD to be able to provision users to <<APPLICATION NAME>> without a pre-integrated provisioning connector.
3. If <<APPLICATION NAME>> is able to utilize the BYOA SCIM connector, then refer to the BYOA SCIM integration [tutorial](https://docs.microsoft.com/azure/active-directory/active-directory-scim-provisioning#getting-started) to configure the BYOA SCIM connector for <<APPLICATION NAME>>.
4. Customize your desired user and/or group attribute mappings for <<APPLICATION NAME>> per the [instructions here](https://docs.microsoft.com/azure/active-directory/active-directory-saas-customizing-attribute-mappings).
5. If the data values between your source and target systems are incompatible, you can [configure expressions for attribute mappings](https://docs.microsoft.com/azure/active-directory/active-directory-saas-writing-expressions-for-attribute-mappings) that will convert your users and/or groups data into formats that are more acceptable for <<APPLICATION NAME>>.
6. Configure the desired users and/or groups that you would like to synchronize to <<APPLICATION NAME>> using [user and group assignments](https://docs.microsoft.com/azure/active-directory/active-directory-coreapps-assign-user-azure-portal) and/or [scoping filters](https://docs.microsoft.com/azure/active-directory/active-directory-saas-scoping-filters).

## Phase 2: User Acceptance Testing (UAT)

Once you have configured automatic user provisioning for <<APPLICATION NAME>>, you will need to run test cases to verify whether this solution meets your organization’s requirements. These test cases should reflect your Business Use Cases. Use the table below to document your test scenarios along with the expected and actual results:

|  |  |  |
| --- | --- | --- |
| Scenarios | Expected Results | Actual Results |
| *e.g. User is added to a group that is assigned to <<APPLICATION NAME>>.* | *e.g. User object is provisioned in <<APPLICATION NAME>>. User can log into <<APPLICATIO NAME>> and perform the desired actions.* |  |
| *e.g. User is removed from a group that is assigned to <<APPLICATION NAME>>.* | *e.g. User object is deprovisioned in <<APPLICATION NAME>>. User cannot log into <<APPLICATIO NAME>> and perform the desired actions.* |  |
| *e.g. User information is updated in Azure AD through Azure AD Connect or via Graph API.* | *e.g. Updated user information is reflected in <<APPLICATION NAME>> after an incremental sync.* |  |
|  |  |  |

Use the results above to determine how to transition your automatic user provisioning implementation into production based on your [established timelines](#_Timelines). Feel free to extend the table as needed.

## Phase 3: Transitioning into production

Once your testing is complete and successful, move your automatic user provisioning implementation into production by repeating all the steps in Phase 1 to Phase 3 in your production environment.

## Phase 4: Rollback steps

If the automatic user provisioning implementation fails to work as desired in the production environment, the following rollback steps below can assist you in reverting back to a previous known good state:

* 1. Review the [provisioning summary report](https://docs.microsoft.com/azure/active-directory/active-directory-saas-provisioning-reporting#getting-provisioning-reports-from-the-azure-management-portal) and [provisioning audit logs](https://docs.microsoft.com/azure/active-directory/active-directory-saas-provisioning-reporting#provisioning-audit-logs) to determine what the incorrect operations were performed on the affected users and/or groups.
  2. The last known good state of the users and/or groups affected can be determined through the provisioning audit logs or by reviewing the source systems (Azure AD or AD).
  3. Work with the application owner to update the users and/or groups affected directly in <<APPLICATION NAME>> using the last known good state values.

# Operationalize your Implementation

This section will guide you in best practices to maintain the automatic user provisioning implementation that has been deployed.

## Reporting and monitoring

Azure AD can provide additional insights into your organization’s user provisioning usage and operational health through audit logs and reports. The table below lists the provisioning reports and logs available along with the insights that they provide:

|  |  |  |
| --- | --- | --- |
| Report Type | Insights | Location |
| Provisioning summary report | * The total number of users and/groups that have been synchronized and are currently in scope for provisioning. * The last time the synchronization was run which typically occur every 20-40 minutes, after a full synchronization has completed. * Determine if an initial full synchronization has been completed. * Whether or not the provisioning process has been placed in quarantine, and what the reason for the quarantine status is. | * Azure management portal |
| Provisioning audit logs | * Import events - recorded each time the Azure AD provisioning service retrieves information about an individual user or group. * Synchronization rule events - report on the results of the attribute mapping rules and any configured scoping filters, after user data has been imported. * Export events - recorded each time the Azure AD provisioning service writes a user account or group object to a target system. These events record all user attributes and their values that were written by the Azure AD provisioning service at the time of the event. If there was an error while writing the user account or group object to the target system, it will be displayed here. * Process escrow - occur when the provisioning service encounters a failure while attempting an operation and begins to retry the operation on a back-off interval of time. An "escrow" event is recorded each time a provisioning operation was retired. | * Azure management portal * Audit API |

To learn more about how to navigate the user provisioning reports and audit logs, refer to the [tutorial here](https://docs.microsoft.com/azure/active-directory/active-directory-saas-provisioning-reporting).

* **Microsoft recommends** that you assume ownership of and consume these reports on a regular basis based on your organization’s requirements. Azure AD retains most audit data for 30 days.

## Troubleshooting

To learn more about common issues that affect automatic user provisioning and how to resolve them, refer to the [troubleshooting documentation here](https://docs.microsoft.com/azure/active-directory/active-directory-application-provisioning-content-map). The table below documents additional user provisioning issues that should be considered:

|  |  |  |
| --- | --- | --- |
| Issue | Possible Cause | Recommended Steps |
| User provisioning stopped working despite configuration not being changed since last known good state. | The admin account password for your application in the **Admin Credentials** section may have been changed and/or expired. | * Verify if you are indeed failing on incorrect credentials by reviewing the provisioning audit logs. * Update your admin account password in your application. |
| You have updated the admin account password for your application in the **Admin Credentials** section but have not yet updated the **Security Token**. | * Verify if you are indeed failing on incorrect credentials by reviewing the provisioning audit logs. * Update your Security Token in the Azure AD portal for your application. |

## Reference Documentation

1. [Debug SAML-based SSO](https://docs.microsoft.com/azure/active-directory/develop/active-directory-saml-debugging)
2. [Customizing claim issued in SAML token](https://docs.microsoft.com/azure/active-directory/develop/active-directory-saml-claims-customization)
3. [Single Sign-on SAML protocol](https://docs.microsoft.com/azure/active-directory/develop/active-directory-single-sign-on-protocol-reference)
4. [Single Sign-Out SAML protocol](https://docs.microsoft.com/azure/active-directory/develop/active-directory-single-sign-out-protocol-reference)
5. [Azure AD B2B](https://docs.microsoft.com/azure/active-directory/active-directory-b2b-what-is-azure-ad-b2b) (for external users such as partners and vendors)
6. [Azure AD Conditional Access](https://docs.microsoft.com/azure/active-directory/active-directory-conditional-access-azure-portal)
7. [Azure Identity Protection](https://docs.microsoft.com/azure/active-directory/active-directory-identityprotection)
8. [SSO access](https://docs.microsoft.com/azure/active-directory/active-directory-appssoaccess-whatis)
9. [MFA Conditional Access for SaaS](https://docs.microsoft.com/azure/active-directory/active-directory-playbook-building-blocks#mfa-conditional-access-for-saas-applications)
10. [Configure Token Lifetimes](https://docs.microsoft.com/azure/active-directory/active-directory-configurable-token-lifetimes)
11. [Claim mapping for Apps via PowerShell](https://docs.microsoft.com/azure/active-directory/active-directory-claims-mapping)
12. [Overview of Azure AD automatic user provisioning](https://docs.microsoft.com/azure/active-directory/active-directory-saas-app-provisioning)
13. [Application specific integration tutorials for user provisioning](https://docs.microsoft.com/azure/active-directory/active-directory-saas-tutorial-list)
14. [Azure AD BYOA SCIM user provisioning](https://docs.microsoft.com/azure/active-directory/active-directory-scim-provisioning)

**IMPORTANT NOTICES**

© 2018 Microsoft Corporation.  All rights reserved.  This document is provided "as-is." Information and views expressed in this document, including URL and other Internet Web site 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

**Customize this document**

To customize this document for your use with a specific application, perform a global replace of <<APPLICATION NAME>> with the name of the SaaS application with which you are working with.