Proventeq365

Product Deployment Pre-requisites

Version: 1.0

2 April 2026

Authored by: Proventeq

Status: Released

Proventeq Ltd, Reading Enterprise Centre, Reading, Berkshire RG6 6BU

www.proventeq.com | +44 118 907 9296 | info@proventeq.com

Table of Contents

DOCUMENT CONTROL

Authors List

Date	Author	Version	Change History
2 Apr 2026	Trupti Sarolkar	1.0	

Reviewers List

Name	Position
Miroslav Ligas	Technical Architect

Approvers List

Name	Position
Rakesh Chenchery	CTO

Related Documents

Document Name	Document Path	Comments

1. About This Guide

1.1. Purpose

This guide describes everything you need to prepare before Proventeq deploys Proventeq365 into your Microsoft 365 tenant and Azure environment. Complete all steps in this guide and share the requested details with your Proventeq contact **before** the agreed deployment date.

1.2. Who Should Read This

This guide is intended for:

- **Microsoft 365 / Azure administrators** — managing the tenant, subscription, and identity settings
- **Security administrators** — responsible for Conditional Access, MFA, and identity governance
- **Network administrators** — relevant only if outbound restrictions, proxies, or private networking are in scope

1.3. Scope and Assumptions

- Proventeq365 is deployed into **your** Azure subscription and Microsoft 365 tenant.
- Proventeq deploys a dedicated Azure resource group that you create and control.
- You are responsible for providing the required access and approvals in line with your organization's security policies.
- **Important:** Never send passwords, client secrets, or certificates by email. Share sensitive values only through your approved secure channel.

1.4. Roles and Responsibilities

Area	Your responsibilities (Customer)	Proventeq's responsibilities
Azure subscription	Create the resource group, grant RBAC access to the deployment identity, approve any domain/DNS changes, and optionally register Azure resource providers.	Deploy Azure resources and validate provisioning once access is in place.
Microsoft 365 tenant	Provide tenant admin contacts, create or invite the deployment account, and approve app registration permissions including admin consent.	Configure Proventeq365 integration and complete post-deployment app registration tasks (e.g., redirect URIs and certificates).
Security & governance	Ensure your MFA and Conditional Access policies allow the agreed deployment approach, and approve any exceptions needed.	Provide a least-privilege access recommendation and support access verification.

2. Readiness Checklist

Complete every item in this checklist before the deployment date and confirm status with your Proventeq contact.

#	Checklist item	Status (Yes / No / N/A)	Notes
1	Microsoft 365 licenses confirmed for all impacted users (see 3.1 Licensing)		
2	Deployment mode confirmed — Read-only or Read-write (see 3.2 Deployment Mode)		
3	Archiving enabled in your Microsoft 365 tenant — only if the archiving feature is required (see 3.3. Archiving)		
4	Azure subscription identified for deployment		
5	Dedicated resource group created in the target region (see 4.1 Create a Resource Group)		
6	Deployment identity created or Proventeq guest account invited (see 4.2 Create a Deployment Identity)		
7	Owner role assigned to the deployment identity on the resource group (see 4.3 Grant Access to the Resource Group)		
8	All required Azure resource providers registered (see 4.4 Register Azure Resource Providers)		
9	App registration created and API permissions granted with admin consent (see 5 EntraId App Registartion)		
10	Custom domain name and DNS ownership confirmed — only if a custom domain is required (see 6 Custom Domain and DNS)		

3. Microsoft 365 Prerequisites

3.1. Licensing

Confirm that all users affected by the deployment have the appropriate Microsoft 365 licenses for the workloads in scope — for example: Teams, SharePoint, Microsoft 365 Groups, Planner, or Viva Engage. Check this against your agreed deployment design.

3.2. Deployment Mode

Confirm which mode Proventeq365 will operate in:

Mode	What it does
Read-only	Reports and insights only — no changes made to your environment
Read-write	Reports, insights, and the ability to remediate issues

Your chosen mode affects the permissions required in Section 5.3 .

3.3. Archiving

> **Note:** Only if using the archiving feature.

If your deployment includes archiving, ensure your archiving provider is configured **before** deployment. Supported providers:

- **Wasabi** — Third-party archiving. Ensure this is configured in your environment.
- **Microsoft 365 Archive** — Ensure Microsoft 365 archiving is enabled in your tenant.

4. Azure Prerequisites

Proventeq365 is deployed into a dedicated Azure resource group within your subscription. Work through Sections 4.1 to 4.4 in order.

4.1. Create a Resource Group

What this does: Creates an isolated container in your Azure subscription where all Proventeq365 resources will live.

4.1.1. Steps — Azure Portal

1. Sign in to the Azure Portal (https://portal.azure.com).
2. Search for **Resource groups** and select it.
3. Click **+ Create**.
4. Select your **Subscription**, enter a **Resource group name**, and choose the **Region** that matches where your Microsoft 365 data is stored. To find your M365 data location, visit: https://learn.microsoft.com/en-us/microsoft-365/enterprise/o365-data-locations
5. Click **Review + Create**, then **create**.

4.1.2. Steps — Azure CLI (alternative)

```
az login
az group create --name "<RESOURCE_GROUP_NAME>" --location "<AZURE_REGION>"

Replace <RESOURCE_GROUP_NAME> with your chosen name and <AZURE_REGION> with the
target region (e.g., uksouth).
```

4.2. Create a Deployment Identity

What this does: Provides Proventeq with a secure identity to use during deployment. Choose one of the two options below — agree the approach with your Proventeq contact first.

4.2.1. Option A — Create a dedicated user account in your tenant

Use this option if your policy requires Proventeq to use an internal account.

4.2.1.1. Steps — Azure Portal

1. Sign in to the **Azure Portal** (https://portal.azure.com) and go to **Microsoft Entra ID**.
2. Select **Users** > **+ New user** > **Create new user**.
3. Enter a display name and a user principal name (UPN) — for example, proventeq-deploy@yourdomain.com.
4. Set a strong temporary password, note it down, and share it with Proventeq via your approved secure channel.
5. Select **Create**.

4.2.1.2. Steps — Azure CLI (alternative)

```
az ad user create \
  --display-name "<DISPLAY_NAME>" \
  --user-principal-name "<USERNAME>@<TENANT_DOMAIN>" \
  --password "<TEMPORARY_PASSWORD>"
```

4.2.2. Option B — Invite Proventeq as a guest user

Use this option if your policy permits external guest access.

4.2.2.1. Steps — Azure Portal

1. Sign in to the **Azure Portal** (https://portal.azure.com) and go to **Microsoft Entra ID**.
2. Select **Users** > **+ New user** > **Invite external user**.
3. Enter the Proventeq email address provided by your Proventeq contact.
4. Add a meaningful display name and click **Invite**.

4.2.2.2. Steps — Azure CLI (alternative)

```
az ad user invite \
  --user-email-address "<PROVENTEQ_EMAIL>" \
  --display-name "<DISPLAY_NAME>" \
  --redirect-url "https://portal.azure.com"
```

4.2.3. Security recommendations for the deployment identity

- **Enable MFA** — Strongly recommended. Even if credentials are compromised, MFA prevents unauthorized sign-in.
- **Limit session lifetime** — Configure short session timeouts and disable persistent sessions to reduce the risk of session hijacking.

4.3. Grant Access to the Resource Group (RBAC)

What this does: Gives the deployment identity the permissions it needs to create and manage resources in the resource group.

4.3.1. Steps — Azure Portal

1. Go to the resource group you created in Section 4.1.
2. In the left-hand menu, select **Access control (IAM)**.
3. Click **Add** > **Add role assignment**.
4. On the **Role** tab, search for and select **Owner**, then click **Next**.
5. On the **Members** tab, click **+ Select members**. Search for and select the deployment user or Proventeq guest account, then click **Select** and **Next**.
6. On the **Conditions** tab, select **Allow user to assign all roles**, then click **Next**.
7. On the **Assignment type** tab, set the assignment type to **Active** and the duration to **Permanent**, then click **Next**.
8. Click **Review + assign** to complete the setup.

4.3.2. Steps — Azure CLI (alternative)

```
az login
az role assignment create \
  --assignee "<DEPLOYMENT_IDENTITY_UPN_OR_OBJECT_ID>" \
  --role "Owner" \
  --scope "/subscriptions/<SUBSCRIPTION_ID>/resourceGroups/<RESOURCE_GROUP_NAME>"
```

4.4. Register Azure Resource Providers

What this does: Authorizes your Azure subscription to use the specific Azure services that Proventeq365 requires. This is a one-time step per subscription.

The following resource providers must be registered:

Provider namespace
Microsoft.ManagedIdentity
Microsoft.Network
Microsoft.ContainerService
Microsoft.ContainerRegistry
Microsoft.ContainerInstance

Microsoft.AlertsManagement
Microsoft.AppConfiguration
Microsoft.OperationalInsights
Microsoft.Insights
Microsoft.Storage
Microsoft.Web
Microsoft.KeyVault
Microsoft.Cdn
Microsoft.App
Microsoft.Authorization
Microsoft.Sql
Microsoft.Resources
Microsoft.ServiceBus

4.4.1. Option 1 — Azure Portal

1. Sign in to the **Azure Portal** (https://portal.azure.com).
2. Search for and select **Subscriptions**, then select your target subscription.
3. In the left-hand menu, select **Resource providers**.
4. Search for each provider namespace listed above. If its status is **Not Registered**, select it and click **Register**.
5. Repeat for all providers in the list. Registration usually completes within 1-2 minutes.

4.4.2. Option 2 — Azure CLI (faster for multiple providers)

Run the following commands after signing in (az login) and selecting your target subscription:

```
az provider register --namespace Microsoft.ManagedIdentity
az provider register --namespace Microsoft.Network
az provider register --namespace Microsoft.ContainerService
az provider register --namespace Microsoft.ContainerRegistry
az provider register --namespace Microsoft.ContainerInstance
az provider register --namespace Microsoft.AlertsManagement
az provider register --namespace Microsoft.AppConfiguration
az provider register --namespace Microsoft.OperationalInsights
az provider register --namespace Microsoft.Insights
az provider register --namespace Microsoft.Storage
```

```
az provider register --namespace Microsoft.Web
az provider register --namespace Microsoft.KeyVault
az provider register --namespace Microsoft.Cdn
az provider register --namespace Microsoft.App
az provider register --namespace Microsoft.Authorization
az provider register --namespace Microsoft.Sql
az provider register --namespace Microsoft.Resources
az provider register --namespace Microsoft.ServiceBus
```

To verify that all providers are registered, run:

```
az provider list --query "[?registrationState=='Registered'].namespace" --output
table
```

5. EntraID App Registration (Microsoft 365)

What this does: Registers Proventeq365 as an application in your Microsoft Entra ID tenant and grants it the permissions it needs to access Microsoft 365 services.

> **Note:** You can either follow the steps below or use the script in the **APPENDIX Entra App creation.**
> The specific API permissions depend on your chosen deployment mode (Read-only or Read-write). Proventeq will confirm the exact permissions list before you complete this step.

5.1. Create the App Registration

5.1.1. Steps — Azure Portal

1. Sign in to the **Azure Portal** (https://portal.azure.com) and go to **Microsoft Entra ID**.
2. Select **App registrations** > **+ New registration**.
3. Enter a meaningful name (e.g., Proventeq365), leave the default account type, and click **Register**.
4. Note the **Application (client) ID** and **Directory (tenant) ID** — you will need to share these with Proventeq.

5.1.2. Steps — Azure CLI (alternative)

```
az ad app create \
  --display-name "<APP_NAME>" \
  --sign-in-audience AzureADMyOrg
```

5.2. Assign the Cloud Application Administrator Role

5.2.1. Steps — Azure Portal

1. In **Microsoft Entra ID**, go to **Roles and administrators**.
2. Search for and select **Cloud Application Administrator**.
3. Click **+ Add assignment**.
4. On the **Members** tab, click **Select member(s)** and add the deployment user account created in Section 4.2.
5. On the **Settings** tab, set the assignment type to **Active**, then confirm.

5.2.2. Azure CLI (alternative)

```
az ad app owner add \
  --id <APP_OBJECT_ID> \
  --owner-object-id <USER_OBJECT_ID>
```

5.3. Grant API Permissions

1. In the app registration, go to **Manage** > **API permissions**.
2. Add the required Microsoft Graph and service permissions for your chosen deployment mode:
   - **Read-only mode:** Apply the permissions shown in the Read-only permissions screenshot provided by Proventeq.
   - **Read-write mode:** Apply the permissions shown in the Read-write permissions screenshot provided by Proventeq (this includes additional permissions compared to Read-only).
3. Once all permissions are added, click **Grant Admin Consent for [your organization]** and confirm.

5.3.1. Read-only mode permissions

5.3.1.1. Microsoft Graph (7 permissions)

API / Permission	Type	Description	Admin Consent
Chat.Read.All	Application	Read all chat messages	Yes
Group.Read.All	Application	Read all groups	Yes
Mail.Read	Application	Read mail in all mailboxes	Yes
Mail.ReadBasic.All	Application	Read basic mail in all mailboxes	Yes
Sites.Read.All	Application	Read items in all site collections	Yes
Team.ReadBasic.All	Delegated	Read the names and descriptions of teams	No
User.Read.All	Application	Read all users' full profiles	Yes

5.3.1.2. SharePoint (2 permissions)

API / Permission	Type	Description	Admin Consent
Sites.Read.All	Application	Read items in all site collections	Yes
User.Read.All	Application	Read user profiles	Yes

5.3.2. Read-write mode permissions

5.3.2.1. Microsoft Graph (23 permissions)

API / Permission	Type	Description	Admin Consent
Chat.Create	Application	Create chats	Yes
Chat.Read.All	Application	Read all chat messages	Yes
Directory.ReadWrite.All	Application	Read and write directory data	Yes
Group.Read.All	Application	Read all groups	Yes
Group.ReadWrite.All	Application	Read and write all groups	Yes
InformationProtectionPolicy.Read.All	Application	Read all published labels and label policies	Yes
Mail.Read	Application	Read mail in all mailboxes	Yes
Mail.ReadBasic.All	Application	Read basic mail in all mailboxes	Yes
offline_access	Delegated	Maintain access to data you have given it access to	No
openid	Delegated	Sign users in	No
profile	Delegated	View users' basic profile	No
RecordsManagement.Read.All	Application	Read Records Management configuration, labels	Yes
RecordsManagement.ReadWrite.All	Application	Read and write Records Management configuration	Yes
SensitivityLabels.Read.All	Application	Get labels tenant scope	Yes
Sites.Archive.All	Application	Archive/reactivate Site Collections without a signature	Yes
Sites.Manage.All	Application	Create, edit, and delete items and lists in all sites	Yes
Sites.Read.All	Application	Read items in all site collections	Yes
Sites.ReadWrite.All	Application	Read and write items in all site collections	Yes
Team.ReadBasic.All	Application	Get a list of all teams	Yes
User.Read	Delegated	Sign in and read user profile	No
User.Read.All	Application	Read all users' full profiles	Yes
User.ReadBasic.All	Application	Read all users' basic profiles	Yes
User.ReadWrite.All	Application	Read and write all users' full profiles	Yes

5.3.2.2. SharePoint (2 permissions)

API / Permission	Type	Description	Admin Consent
Sites.FullControl.All	Application	Have full control of all site collections	Yes
User.ReadWrite.All	Application	Read and write user profiles	Yes

6. CUSTOM DOMAIN AND DNS (IF APPLICABLE)

This section applies only if Proventeq365 will use a custom domain name for its endpoints.

1. Confirm the **custom domain name** to be used for Proventeq365 (e.g., proventeq365.yourdomain.com).
2. Identify who manages DNS for this domain within your organization.
3. Confirm the process and approval steps for adding CNAME records — Proventeq will provide the specific values required during deployment.

Share this information with your Proventeq contact before the deployment date.

7. Next Steps

Once all checklist items in 2. Readiness Checklist are complete:

1. Share the completed checklist and all requested details (see section below) with your Proventeq contact.
2. Proventeq will review the information and confirm readiness for deployment.
3. Deployment will proceed on the agreed date.

7.1. Share Details with Proventeq

Once complete, provide your Proventeq contact with:

- The **Application (client) ID** of the Entra Application Registration to access M365
- The **Directory (tenant) ID**
- The **UPN** and **password** of the deployment identity.

Proventeq will complete the remaining post-deployment configuration (such as adding certificates and redirect URIs) as part of the deployment engagement.

If you have any questions or need assistance with any of the steps in this guide, contact your Proventeq project manager or email support@proventeq.com.

8. APPENDIX

8.1. Entra App creation

A complete script to create the Entra app registration and configure it

```
#!/bin/bash
# ------------------------------------------------------------------------------
# Script: create-proventeq-governance-app.sh
# Purpose: Create the Entra ID (Azure AD) App Registration
#          "Proventeq Governance - Combined"
#
# Author: Proventeq
# Audience: Entra ID / Azure AD Global Admin
#
# IMPORTANT LIMITATIONS (BY DESIGN IN MICROSOFT GRAPH):
# - Existing client secret VALUES cannot be recreated
# - Certificate private keys cannot be recreated (public cert only)
# - Original key IDs are always regenerated
#
# This script recreates the APPLICATION CONFIGURATION ONLY.
# ------------------------------------------------------------------------------

set -e

# ------------------------------------------------------------------------------
# 1. PREREQUISITES
# ------------------------------------------------------------------------------
# You must be logged in with sufficient privileges:
# - Application.ReadWrite.All
# - Directory.ReadWrite.All
# - Global Administrator (for admin-consent)
#
# Login:
#   az login
#
# Optional: Lock to correct tenant/subscription
#   az account show
# ------------------------------------------------------------------------------


# ------------------------------------------------------------------------------
# 2. VARIABLES (SAFE TO EDIT)
# ------------------------------------------------------------------------------

read -rp "Enter App Registration display name: " APP_NAME

echo ""
echo "Select permission mode:"
echo "  1) Read-only"
echo "  2) Read-write"
read -rp "Enter choice (1 or 2): " PERM_MODE_CHOICE

case "$PERM_MODE_CHOICE" in
  1) PERM_MODE="readonly"  ;;
```

```
  2) PERM_MODE="readwrite" ;;
  *) echo "Invalid choice. Defaulting to read-only."; PERM_MODE="readonly" ;;
esac

echo "  Permission mode: $PERM_MODE"

SIGN_IN_AUDIENCE="AzureADMyOrg"

# OAuth2 scope ID (must be stable if relied upon by clients)
OAUTH_SCOPE_ID="b9e69f89-9f2f-4f3a-bcb5-e5f9b03e052e"


# ------------------------------------------------------------------------------
# 3. CHECK LOGIN
# ------------------------------------------------------------------------------

echo ""
echo "Checking Azure login status..."

if ! az account show > /dev/null 2>&1; then
    echo "Not logged in. Running az login..."
    az login
fi

TENANT_ID=$(az account show --query tenantId -o tsv)
USER_NAME=$(az account show --query user.name -o tsv)
SUB_NAME=$(az account show --query name -o tsv)

echo "  Logged in as : $USER_NAME"
echo "  Tenant       : $TENANT_ID"
echo "  Subscription : $SUB_NAME"


# ------------------------------------------------------------------------------
# 4. CREATE APP REGISTRATION
# ------------------------------------------------------------------------------
# Creates the app WITHOUT identifierUris first, then sets api://<appId>
# (custom domains like gov360.proventeq.com are not verified in external tenants)
# ------------------------------------------------------------------------------

echo ""
echo "Step 1: Creating App Registration..."

BODY_FILE=$(mktemp)
cat > "$BODY_FILE" <<EOF
{
  "displayName": "$APP_NAME",
  "signInAudience": "$SIGN_IN_AUDIENCE",
  "spa": {
    "redirectUris": []
  },
  "api": {
    "oauth2PermissionScopes": [
      {
        "id": "$OAUTH_SCOPE_ID",
        "type": "Admin",
        "value": "access_as_user",
```

```
        "adminConsentDisplayName": "access_as_user",
        "adminConsentDescription": "Access the application",
        "isEnabled": true
      }
    ]
  }
}
EOF

APP_OBJECT_ID=$(az rest \
  --method POST \
  --url https://graph.microsoft.com/v1.0/applications \
  --headers Content-Type=application/json \
  --body @"$BODY_FILE" \
  --query id -o tsv)

rm -f "$BODY_FILE"

if [ -z "$APP_OBJECT_ID" ]; then
    echo "ERROR: Failed to create app registration."
    exit 1
fi

APP_ID=$(az rest \
  --method GET \
  --url "https://graph.microsoft.com/v1.0/applications/$APP_OBJECT_ID" \
  --query appId -o tsv)

echo "  App created successfully."
echo "  App Object ID : $APP_OBJECT_ID"
echo "  App Client ID : $APP_ID"

# Step 1b: Set identifierUris using api://<appId> (works in any tenant)
echo "  Setting Identifier URI..."
IDENTIFIER_URI_FINAL="api://$APP_ID"

URI_FILE=$(mktemp)
cat > "$URI_FILE" <<EOF
{
  "identifierUris": ["$IDENTIFIER_URI_FINAL"]
}
EOF

az rest \
  --method PATCH \
  --url "https://graph.microsoft.com/v1.0/applications/$APP_OBJECT_ID" \
  --headers Content-Type=application/json \
  --body @"$URI_FILE"

rm -f "$URI_FILE"

echo "  Identifier URI: $IDENTIFIER_URI_FINAL"


# ------------------------------------------------------------------------------
# 5. ADD REQUIRED API PERMISSIONS
# ------------------------------------------------------------------------------
```

```
# Permissions applied depend on the selected mode:
#
# READ-ONLY
#   SharePoint : Sites.Read.All, User.Read.All
#   Graph      : Chat.Read.All, Group.Read.All, Mail.Read, Mail.ReadBasic.All,
#                Sites.Read.All, Team.ReadBasic.All (delegated), User.Read.All
#
# READ-WRITE
#   SharePoint : Sites.FullControl.All, User.ReadWrite.All
#   Graph      : Chat.Create, Chat.Read.All, Directory.ReadWrite.All,
#                Group.Read.All, Group.ReadWrite.All,
#                InformationProtectionPolicy.Read.All, Mail.Read,
#                Mail.ReadBasic.All, offline_access, openid, profile,
#                RecordsManagement.Read.All, RecordsManagement.ReadWrite.All,
#                SensitivityLabels.Read.All, Sites.Archive.All,
#                Sites.Manage.All, Sites.Read.All, Sites.ReadWrite.All,
#                Team.ReadBasic.All (application), User.Read,
#                User.Read.All, User.ReadBasic.All, User.ReadWrite.All
#
# NOTE: This does NOT automatically grant admin consent.
# ------------------------------------------------------------------------

echo ""
echo "Step 2: Configuring required API permissions ($PERM_MODE)..."

PERM_FILE=$(mktemp)

if [ "$PERM_MODE" = "readonly" ]; then
cat > "$PERM_FILE" <<'EOF'
{
  "requiredResourceAccess": [
    {
      "resourceAppId": "00000003-0000-0ff1-ce00-000000000000",
      "resourceAccess": [
        { "id": "4e0d77b0-96ba-4398-af14-3baa780278f4", "type": "Role" },
        { "id": "56680e0d-d2a3-4ae1-80d8-3c4a5f90e2bf", "type": "Role" }
      ]
    },
    {
      "resourceAppId": "00000003-0000-0000-c000-000000000000",
      "resourceAccess": [
        { "id": "6b7d71aa-70aa-4810-a8d9-5d9fb2830017", "type": "Role" },
        { "id": "5b567255-7703-4780-807c-7be8301ae99b", "type": "Role" },
        { "id": "810c84a8-4a9e-49e6-bf7d-12d183f40d01", "type": "Role" },
        { "id": "693c5e45-0940-467d-9b8a-1022fb9d42ef", "type": "Role" },
        { "id": "332a536c-c7ef-4017-ab91-336970924f0d", "type": "Role" },
        { "id": "485be79e-c497-4b35-9400-0e3fa7f2a5d4", "type": "Scope" },
        { "id": "df021288-bdef-4463-88db-98f22de89214", "type": "Role" }
      ]
    }
  ]
}
EOF
else
cat > "$PERM_FILE" <<'EOF'
{
  "requiredResourceAccess": [
```

```
    {
      "resourceAppId": "00000003-0000-0ff1-ce00-000000000000",
      "resourceAccess": [
        { "id": "678536fe-1083-478a-9c59-b99265e6b0d3", "type": "Role" },
        { "id": "741f803b-c850-494e-b5df-cde7c675a1ca", "type": "Role" }
      ]
    },
    {
      "resourceAppId": "00000003-0000-0000-c000-000000000000",
      "resourceAccess": [
        { "id": "d9c48af6-9ad9-47ad-82c3-63757137b9af", "type": "Role" },
        { "id": "6b7d71aa-70aa-4810-a8d9-5d9fb2830017", "type": "Role" },
        { "id": "19dbc75e-c2e2-444c-a770-ec69d8559fc7", "type": "Role" },
        { "id": "5b567255-7703-4780-807c-7be8301ae99b", "type": "Role" },
        { "id": "62a82d76-70ea-41e2-9197-370581804d09", "type": "Role" },
        { "id": "19da66cb-0fb0-4390-b071-ebc76a349482", "type": "Role" },
        { "id": "810c84a8-4a9e-49e6-bf7d-12d183f40d01", "type": "Role" },
        { "id": "693c5e45-0940-467d-9b8a-1022fb9d42ef", "type": "Role" },
        { "id": "7427e0e9-2fba-42fe-b0c0-848c9e6a8182", "type": "Scope" },
        { "id": "37f7f235-527c-4136-accd-4a02d197296e", "type": "Scope" },
        { "id": "14dad69e-099b-42c9-810b-d002981feec1", "type": "Scope" },
        { "id": "ac3a2b8e-03a3-4da9-9ce0-cbe28bf1accd", "type": "Role" },
        { "id": "eb158f57-df43-4751-8b21-b8932adb3d34", "type": "Role" },
        { "id": "e46a01e9-b2cf-4d89-8424-bcdc6dd445ab", "type": "Role" },
        { "id": "e3530185-4080-478c-a4ab-39322704df58", "type": "Role" },
        { "id": "0c0bf378-bf22-4481-8f81-9e89a9b4960a", "type": "Role" },
        { "id": "332a536c-c7ef-4017-ab91-336970924f0d", "type": "Role" },
        { "id": "9492366f-7969-46a4-8d15-ed1a20078fff", "type": "Role" },
        { "id": "2280dda6-0bfd-44ee-a2f4-cb867cfc4c1e", "type": "Role" },
        { "id": "e1fe6dd8-ba31-4d61-89e7-88639da4683d", "type": "Scope" },
        { "id": "df021288-bdef-4463-88db-98f22de89214", "type": "Role" },
        { "id": "97235f07-e226-4f63-ace3-39588e11d3a1", "type": "Role" },
        { "id": "741f803b-c850-494e-b5df-cde7c675a1ca", "type": "Role" }
      ]
    }
  ]
}
EOF
fi

az rest \
  --method PATCH \
  --url "https://graph.microsoft.com/v1.0/applications/$APP_OBJECT_ID" \
  --headers Content-Type=application/json \
  --body @"$PERM_FILE"

rm -f "$PERM_FILE"

echo "  API permissions configured."

# ------------------------------------------------------------------------------
# 6. CREATE SERVICE PRINCIPAL
# ------------------------------------------------------------------------------
# Required for:
# - API permission grants
# - Enterprise Applications view
# ------------------------------------------------------------------------------
```

```
echo ""
echo "Step 3: Creating Service Principal..."

SP_FILE=$(mktemp)
cat > "$SP_FILE" <<EOF
{
  "appId": "$APP_ID"
}
EOF

az rest \
  --method POST \
  --url https://graph.microsoft.com/v1.0/servicePrincipals \
  --headers Content-Type=application/json \
  --body @"$SP_FILE"

rm -f "$SP_FILE"

echo "  Service Principal created."


# ------------------------------------------------------------------------------
# 7. ADMIN CONSENT (OPTIONAL BUT RECOMMENDED)
# ------------------------------------------------------------------------------
# Grants consent for all delegated/application permissions
# Requires Global Admin
# ------------------------------------------------------------------------------

echo ""
echo "Step 4: Granting admin consent..."
az ad app permission admin-consent --id "$APP_ID"
echo "  Admin consent granted."


# ------------------------------------------------------------------------------
# 8. CREDENTIALS (MANUAL / OPTIONAL)
# ------------------------------------------------------------------------------
# Client secrets and certificates must be created separately.
#
# Example - CREATE A NEW CLIENT SECRET:
#
# az ad app credential reset \
#   --id $APP_OBJECT_ID \
#   --display-name "Prod Env" \
#   --years 1
#
# Example - ADD CERTIFICATE (PUBLIC KEY ONLY):
#
# az ad app credential reset \
#   --id $APP_OBJECT_ID \
#   --cert @proventeq365.cer \
#   --append
#
# ------------------------------------------------------------------------------
```

```
# ------------------------------------------------------------------------------
# 9. FINAL OUTPUT
# ------------------------------------------------------------------------------

echo ""
echo "------------------------------------------------------"
echo "App Registration Complete"
echo "Display Name   : $APP_NAME"
echo "App (client) ID: $APP_ID"
echo "Object ID      : $APP_OBJECT_ID"
echo "Identifier URI : $IDENTIFIER_URI_FINAL"
echo "------------------------------------------------------"
```