

AEU55 – AI Generated Meeting summary

Quick recap

This was the September Access Europe User Group meeting where Peter Bryant and Andrew Richards presented their Graph Authenticator tool, which enables Microsoft Access to connect with Microsoft 365 services including Outlook, Teams, SharePoint, and other APIs. The presenters explained how the tool works by creating Azure app registrations with proper permissions, obtaining bearer tokens for authentication, and connecting to various Microsoft 365 endpoints. They demonstrated the technical process of setting up an app in the Azure portal, including creating app registrations, adding API permissions, and managing client IDs and tenant IDs. The session covered how the tool can handle multi-factor authentication and work with different Microsoft services, with Peter handling the technical demonstrations while Andrew provided context about the tool's development history and capabilities.

Graph Authenticator Microsoft 365 Integration

Andrew and Peter presented their Microsoft 365 integration tool called Graph Authenticator, which addresses the issue of new Outlook not supporting VBA automation. They explained that the tool allows Access to connect to various Microsoft 365 services including Teams, Intune, SharePoint, and email systems, providing full functionality despite the limitations of new Outlook. The presentation included background on the 12-15 month development process and demonstrated the tool's capabilities for connecting to Microsoft services through Entra apps with defined scopes and permissions.

Azure App Registration Configuration Process

Andrew explained the process of creating and configuring Azure app registrations, including the use of tenant IDs, client IDs, URL endpoints, and bearer tokens for authentication and authorization. Peter demonstrated the step-by-step process of creating an app registration in the Azure portal, setting permissions, and highlighting the differences between application and delegated permissions. Andrew emphasized the importance of carefully managing app scopes and permissions to prevent unauthorized access, and mentioned that their sample database includes code examples for creating apps and performing actions like sending emails.

Microsoft Entra API Automation Tool

Peter demonstrated a database tool that automates the creation of 27 different API applications in Microsoft Entra, showing the process of configuring permissions and handling authentication with 2FA. The tool builds URLs and handles API permissions automatically, reducing manual programming effort from two weeks to an automated process. During the demonstration, Peter successfully created multiple app registrations and tested email functionality, with Andrew receiving a test email as confirmation.

SendMail Capabilities and Technical Demo

Peter demonstrated the capabilities of Mail.Send, confirming that it supports group email sending through the "send on behalf as" feature, provided the user has appropriate permissions. He clarified that 27 different apps were created, each with unique capabilities rather than a single app with multiple permissions. The discussion covered technical aspects of accessing Microsoft Graph API, including the process of connecting to Azure apps, acquiring bearer tokens, and sending emails through HTTP requests. Peter also mentioned the introduction of four new scopes in Entra by Microsoft and discussed caching mechanisms to handle authentication in the demo database.

Microsoft 365 Authentication Caching System

Peter demonstrated a new caching system for Microsoft 365 authentication that allows users to authenticate once every 60-90 minutes across multiple applications rather than repeatedly. The system stores tokens in

memory and can programmatically control authentication prompts by consolidating features into monolithic apps with combined permissions. Version 1.1 will include refresh token capabilities to automatically extend token validity without user interaction, providing a more seamless experience similar to Outlook.

Graph Authenticator Tool Demonstration

Andrew and Peter demonstrated Graph Authenticator, a tool that allows developers to authenticate with Microsoft 365 APIs. They explained how authentication prompts work, including when tokens expire and why multiple prompts occur during demo testing. The tool is available as a DLL with different licensing options: a developer license for £445 (with a conference discount using code THANKSCOLIN), a 12-month license for £195, and a deployment license sold to customers for £995. The tool provides comprehensive documentation and is being used by various developers in production environments.

Multi-App vs Single App Strategy

The discussion focused on the pros and cons of building multiple single-scope apps versus a single multi-scope app, with Andrew suggesting that a single app approach might be preferable for replacing Outlook functionality, though there could be cases where limited permissions are needed. The group discussed Microsoft's decision to make email automation more complex, which Andrew attributed to Microsoft's broader push toward web-based solutions rather than desktop applications. The conversation also covered authentication requirements, where Peter explained that re-authentication would be needed each time Access is closed and reopened, though there are potential security risks associated with preserving authentication tokens. When asked about intercepting COM object send commands, Peter clarified that their approach replaces the existing code rather than intercepting it, demonstrating how they would handle report creation and email sending through VBAAuth.

Email Button and Token Solutions

The team discussed solutions for handling the email button in print preview, with Andrew and Colin agreeing that removing the built-in button and creating a custom ribbon using tools like Kevin Bell's would be the best approach. Neil raised questions about token storage, specifically asking about saving refresh tokens in Credential Manager, to which Peter explained the current approach of using MSAL libraries for token management and the potential for future implementation in version 1.1. Stephen inquired about the license checking tool for developers, which Peter described as a simple application for dragging and dropping license files. Marcus expressed appreciation for the tool and shared his experience creating a similar interface for email and calendar functions with a VBA interface.

OAuth Authentication Implementation Discussion

The team discussed the implementation of OAuth authentication for Microsoft Graph API, with Andrew and Peter explaining their decision to use a DLL approach rather than browser-based authentication due to Microsoft's recommendations against URL-based token handling methods. Neil questioned why MSAL was required instead of standard OAuth, but the team clarified that their approach was chosen to ensure robust, officially supported authentication for client deployments.