

Migrate Local User Profile to Entra ID (Intune-Joined) Profile Using USMT

When moving a local Windows user profile to an **Entra ID (Azure AD) joined profile**, you must ensure that **user data, app settings, and files** are migrated correctly. USMT (User State Migration Tool) helps achieve this.

Step 1: Install USMT on the Source Device

USMT is included in the Windows Assessment and Deployment Kit (Windows ADK).

<https://learn.microsoft.com/en-us/windows-hardware/get-started/adk-install#download-the-adk-101261002454-december-2024>

Install Deployment Tools from ADK.

Install User State Migration Tool (USMT) from Windows ADK Add-on.

Locate USMT Executables

USMT files are stored at:

C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\User State Migration Tool\amd64

Step 2: Create a Migration Store Using ScanState

```
cd "C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\User State Migration Tool\amd64"
```

```
scanstate.exe C:\MigStore /i:migdocs.xml /i:migapp.xml /o /c /v:13  
/l:C:\MigStore\ScanState.log
```

Explanation of Parameters:

Parameter	Description
C:\MigStore	Location where user data will be stored
/i:migdocs.xml	Captures user documents and files
/i:migapp.xml	Captures app settings
/o	Overwrites existing store (if applicable)
/c	Continues even if errors occur
/v:13	Sets verbose logging to level 13 (detailed logs)
/l:C:\MigStore\ScanState.log	Saves log file for troubleshooting

Step 3: Ensure the Entra ID User Profile Exists

Before restoring data, **log into the target device** using the Entra ID user.

Run in **Command Prompt** to verify:

```
whoami /upn
```

It should return something like:

```
AzureAD\atQoradmin@yourdomain.com
```

If the user has **not signed in**, ask them to do so first.

Step 4: Restore Data to the Entra ID User Profile

Use the loadstate command to migrate data

```
cd "C:\Program Files (x86)\Windows Kits\10\Assessment and Deployment Kit\User  
State Migration Tool\amd64"
```

```
loadstate.exe C:\MigStore /i:migdocs.xml /i:migapp.xml /mu:Windows10-  
VM\admini:AzureAD\atQoradmin@yourdomain.com /c /lac /v:13  
/l:C:\MigStore\LoadState.log
```

Explanation of Parameters:

Parameter	Description
C:\MigStore	Location of the saved migration store
/i:migdocs.xml	Restores user documents and files
/i:migapp.xml	Restores app settings
/mu:Windows10-VM\admini:AzureAD\atQoradmin@yourdomain.com	Maps local user to Entra ID user
/c	Continues even if errors occur
/lac	Creates the Entra ID user if not found
/v:13	Sets verbose logging for troubleshooting
/l:C:\MigStore\LoadState.log	Saves the log file

Step 5: Verify Migration

Check if the user profile is migrated:

- Go to C:\Users\IshankKansara
- Ensure files and settings are intact

Check event logs for errors:

- Open Event Viewer (eventvwr.msc)
- Navigate to Applications and Services Logs → Microsoft → Windows → User Profile Service

Check USMT logs for errors:

notepad C:\MigStore\LoadState.log

Common Issues & Fixes

Error: "No mapping between account names and security IDs (1332)"

Fix:

- Ensure the user has **logged into the device at least once**.

Error: "Read error 13: Data is invalid"

Fix:

- Check if the **USMT.MIG file is corrupted**.
- Re-run scanstate.exe with /c to ignore non-critical errors

Application Settings Covered by USMT

USMT primarily migrates user-based application settings rather than the application itself. The settings that get migrated depend on the XML configuration files used (migapp.xml, migdocs.xml, or custom XML files). Below is an overview of the **application settings USMT can migrate**:

Microsoft Office Settings

- ✓ Word, Excel, PowerPoint, Outlook, OneNote
- ✓ Custom templates, macros, and settings
- ✓ Outlook profiles, signatures, and email settings

Example:

- If a user has custom ribbon settings in Word or an Outlook signature, those settings will be migrated.

Web Browsers & Internet Settings

- ✓ Internet Explorer (IE) Favorites, Cookies, and History
- ✓ Edge, Chrome, and Firefox bookmarks (if configured in XML)
- ✓ Proxy settings and homepage preferences

Example:

- If a user has bookmarks in Chrome, these can be migrated by modifying migapp.xml to include Chrome's profile data.

Third-Party Application Settings

USMT can migrate some **third-party application settings**, but support is limited unless custom XML files are used.

- ✓ Adobe Photoshop preferences
- ✓ Visual Studio settings
- ✓ AutoCAD user preferences
- ✓ SQL Management Studio settings

Example:

- If a developer has configured Visual Studio with a specific theme, USMT can migrate those settings.

Custom File Type Associations

Default programs for opening files (e.g., PDFs opening with Adobe Reader instead of Edge)

Example:

- If a user prefers VLC as the default media player, USMT can migrate that preference.

What is NOT Migrated?

Installed applications (USMT **only migrates settings**, not the software itself)

User licenses or product keys (e.g., Office activation keys)

Encrypted files unless the encryption key is also migrated

Some browser extensions and plugins

Windows Store (UWP) apps

How to Include More Application Settings?

If you want to migrate settings from additional applications, you can create a **custom XML file** to include them.

Example: Custom XML for Chrome Migration

```
<migration>
  <component type="Application" context="User">
    <displayName>Google Chrome</displayName>
    <rules>
      <include>
        <objectSet>
          <pattern>%LOCALAPPDATA%\Google\Chrome\User Data\Default</pattern>
        </objectSet>
      </include>
    </rules>
  </component>
</migration>
```

Save it as **ChromeSettings.xml** and include it in the scanstate and loadstate commands:

```
scanstate C:\MigStore /i:migdocs.xml /i:migapp.xml /i:ChromeSettings.xml /o /c /v:13
/l:C:\MigStore\ScanState.log
```

```
loadstate C:\MigStore /i:migdocs.xml /i:migapp.xml /i:ChromeSettings.xml /c /v:13
/l:C:\MigStore\LoadState.log
```