Intune Backup and Restore
Script Operations
Revision A

ACHIEVE ONE

Table of Contents

Document Purpose

This document provides comprehensive guidance for operating the IntuneBackupRestore script, which facilitates automated backup and single-file restore of Microsoft Intune configurations. It outlines the necessary prerequisites, including required software and modules, and details the steps to verify and prepare the environment. Instructions are included for performing backups with optional retention policies, restoring configurations from a single JSON file, and ensuring data integrity through unique naming conventions. Additionally, the document explains how to configure a scheduled task for recurring nightly backups, including secure handling of credentials using DPAPI. Recovery procedures and logging mechanisms are also covered to support troubleshooting and auditability.

DPAPI Usage for Secrets

The script supports secure storage of the **Client Secret** using **Windows Data Protection API (DPAPI)**. When creating a scheduled task via the UI, the secret is encrypted and saved to a file (e.g., C:\ProgramData\IntuneBR\secret.bin) using machine-level protection. This ensures:

- The secret is never stored in plain text in configuration files.
- Only the same machine can decrypt the secret.
- The scheduled task uses the encrypted secret for unattended backups without exposing credentials.

Dry Run vs Force Restore

- **Dry Run**
  When the **Dry Run** option is selected (or -ForceRestore is omitted in CLI), the script simulates the restore process without making any changes in Microsoft Intune. It validates the input JSON, checks the target URI, and logs what would happen if the restore were executed. This mode is helpful for:
  - Verifying the integrity of the backup file.
  - Confirming that the restore operation will succeed without errors.
  - Reviewing the planned changes before committing.

- **Force Restore**
  When **Dry Run** is unchecked in the UI or the -ForceRestore switch is included in the command line, the script performs an actual restore by sending a **POST** request to Microsoft Graph. This creates a new configuration object in Intune based on the JSON file, ensuring:
  - No existing configuration is overwritten.
  - A unique name is applied (with timestamp if -AppendTimestamp is used) to avoid conflicts.

Dry Run Best Practice:

Always run a **Dry Run** first to confirm the restore plan, then proceed with **Force Restore** once validated.

Document Change History

Author/Contributor	Version	Date	Description of Changes
Michael Molle	--	10/28/25	Initial Release

Prerequisites

1

PowerShell Module Installation

The Intune backup and Restore script has prerequisites for proper operation. These can be installed before executing the script using a companion script called Pre-InstallPrerequisites.ps1, or, when the script executes, it will detect prerequisites and attempt to install them.

- PowerShell Version
  - PowerShell 7.x (recommended)
  - Compatible with Windows PowerShell 5.1, but some UI features require 7.x.
- .NET Runtime
  - .NET Windows Desktop Runtime 8 or higher (required for WPF-based UI mode).
- Microsoft Graph PowerShell SDK
  - Recommend ensuring the latest version is running before execution
- Companion Script
  - A helper script named Pre-Install-Prerequisites.ps1 is available in the same GitHub
  - Run this script before using IntuneBackupRestore to install and configure all prerequisites automatically.

Configuring Execution Policy and Unblocking the Script

To ensure the script runs without security restrictions, you may need to adjust the PowerShell execution policy and unblock the downloaded file.

Set Execution Policy
This allows locally created scripts to run and requires downloaded scripts to be signed.

Run PowerShell as Administrator and execute:

- Set-ExecutionPolicy RemoteSigned -Scope CurrentUser

Unblock the Script File
If the script was downloaded from the internet, Windows may block it. To unblock:

- Unblock-File -Path .\IntuneBackupRestore.beta.1.1.9.ps1

Alternatively, you can verify if the file is blocked:

- Get-Item .\IntuneBackupRestore.beta.1.1.9.ps1 | Select-Object -Property Name, IsReadOnly, Attributes

Backing up Intune

2

Backup Procedure

Follow these steps to perform a backup using the IntuneBackupRestore script:

Parameters explained:

-TenantId and -AppId: Your Azure AD application credentials.
-ClientSecretPlain: The client secret for authentication.
-OutputPath: Directory where backups will be stored.
-UseTimestampFolder: Creates a timestamped folder for each backup.
-RetentionCount: Number of backup folders to retain (older ones are deleted).

Optional Switches:

-IncludeManagedDevices: Include device inventory (slower).
-AutoBeta: Automatically use beta Graph endpoints for certain entities.
-Diag: Enable diagnostic logging.

Command Line Backup

- Launch PowerShell
    - Use PowerShell 7.x for best compatibility.
- Navigate to the directory containing the script.
- Run in Backup Mode
- Execute the script with required parameters:
    - PowerShellpwsh -NoProfile -ExecutionPolicy Bypass -File .\IntuneBackupRestore.beta.1.1.9.ps1 -Mode Backup -TenantId "<YourTenantId>" -AppId "<YourAppId>" -ClientSecretPlain "<YourClientSecret>" -OutputPath "C:\Staging\Backup" -UseTimestampFolder -RetentionCount 10Show more lines

Verify Backup

-After completion, check the output directory for timestamped folders.
-Review the log file located in the backup folder for details.

Using the UI for Backup

The script provides a graphical user interface (WPF) for users who prefer a point-and-click experience. To launch the UI:

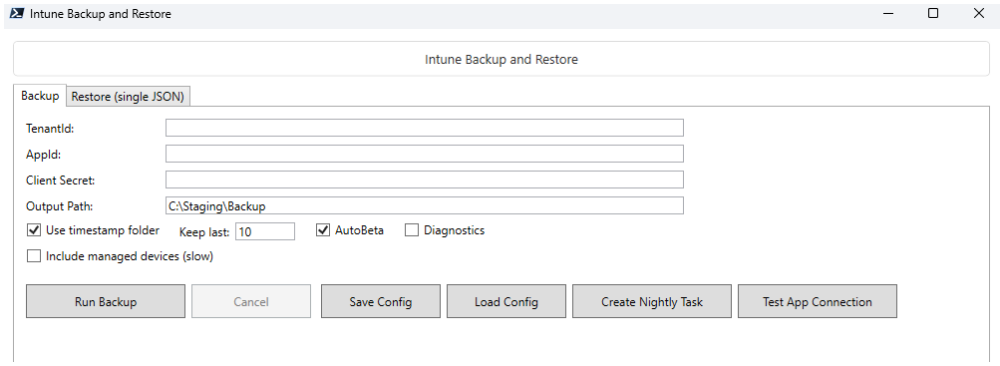

Figure 1-UI Mode

1. **Start the Script in UI Mode**
   - pwsh -NoProfile -ExecutionPolicy Bypass -File .\IntuneBackupRestore.beta.1.1.9.ps1 -Mode UI
2. **Fill Required Fields**
   - **TenantId**: Your Azure AD tenant ID.
   - **AppId**: The application (client) ID.
   - **Client Secret**: Enter the app secret securely.
   - **Output Path**: Directory for storing backups.
   - Optional: Enable **Use timestamp folder**, set **Retention count**, and toggle **Include managed devices** if needed.
3. **Run Backup**
   - Click **Run Backup** to start the process.
   - Monitor status in the UI; a link to open the backup folder will appear when complete.
4. **Save Configuration**
   - Click **Save Config** to export your current settings (excluding secrets) to a JSON file.
   - Example file name: IntuneBR.config.json.
5. **Load Configuration**
   - Click **Load Config** to import previously saved settings.
   - This restores all non-secret fields for quick setup.

Restore Individual Profile to Intune

3

Restore Procedure

Using the UI

1. Launch the script in **UI mode**:
    - pwsh -File .\IntuneBackupRestore.ps1 -Mode UI
2. Navigate to the **Restore (single JSON)** tab.
    - Enter:
        1. **Tenant ID**
        2. **App ID**
        3. **Client Secret**
3. Browse and select the JSON file to restore.
4. Choose options:
    - **Dry Run** (plan only, no changes)
    - **Append Timestamp** (recommended to avoid name conflicts)
5. Click **Run Restore**.
    - If **Dry Run** is unchecked, the script will POST the configuration to Microsoft Graph.

Using Command Line

1. From Admin Powershell Run:
    - pwsh -File .\IntuneBackupRestore.ps1 -Mode Restore `
      -TenantId "<TenantID>" `
      -AppId "<AppID>" `
      -ClientSecretPlain "<Secret>" `
      -InputPath "C:\Backup\deviceConfigurations\Policy.json" `
      -ForceRestore `
      -AppendTimestamp

Notes

- -ForceRestore commits changes.
- -AppendTimestamp ensures unique naming.

Recurring Backup with Schedule Task

4

Creating a Scheduled Task via UI

The script includes a Create Nightly Task button in its UI, which automates the setup of a recurring backup process. When clicked, it securely stores the client secret using DPAPI encryption and registers a Windows Scheduled Task that runs the backup mode daily at 2:00 AM. The task uses PowerShell with the same parameters configured in the UI, ensuring backups are performed without manual intervention. This feature simplifies long-term maintenance by ensuring consistent backups and retention management in accordance with the settings you define.

Automated Creation of a Scheduled Task

1. In the Backup tab, fill in:
    - Tenant ID, App ID, Client Secret, Output Path.
2. Click Create Nightly Task.
    - The script will:
        - Encrypt the secret using DPAPI.
        - Register a scheduled task named IntuneBR Nightly Backup.
        - Configure it to run daily at 2:00 AM with hidden execution.
3. Verify the task in Task Scheduler under Task Scheduler Library.

Edit Using Task Scheduler GUI

- Open Task Scheduler (taskschd.msc).
- Navigate to Task Scheduler Library → locate the task named IntuneBR Nightly Backup.
- Right-click the task and select Properties.
- From here, you can:
    - Change the trigger time (e.g., from 2:00 AM to another time).
    - Adjust conditions (e.g., run only if on AC power).
    - Modify settings like “Run whether user is logged on or not” or “Stop the task if it runs longer than X hours.”