BIG-IP® APM and F5 Access for MacOS

Version 2.0.0

Table of Contents

Overview: F5 Access for macOS Devices

F5 Access for macOS general information

General F5 Access Information

F5 Access for macOS provides Layer 3 network access for the BIG-IP APM module. The F5 Access for macOS SSL VPN application complements the existing Edge Client VPN product line, addressing similar use-case and deployment scenarios.

F5 Access for macOS incorporates Apple's new Network Extension Framework. This change creates some major architectural shifts in the new F5 Access VPN application. As a result, there are currently feature differences between F5 Access and Edge Client for macOS.

Note: *Users can install and use both F5 Access and Edge Client for macOS on the same system.*

Self-signed BIG-IP certificates are not supported unless the CA certificate is first Trusted on the device. Set the system keychain settings to **Always Trust**.

Note: *F5 Access for macOS is hosted in the Apple App Store, instead of on a BIG-IP system.*

F5 Access for macOS has two components:

- **App Extension**: built on the Network Extension framework to provide traffic tunneling.
- **F5 Access Container App**: handles configuration management and state monitoring.

Supported Authentication Modes

Native

Native authentication mode is the default mode that the administrator can use to set the user logon by using username and password, optional client certificate, or both. Interactive authentication, including SAML and external logon pages, are not supported in this mode. Native mode does not require user interaction if all the credentials are previously saved.

Web (Web Logon)

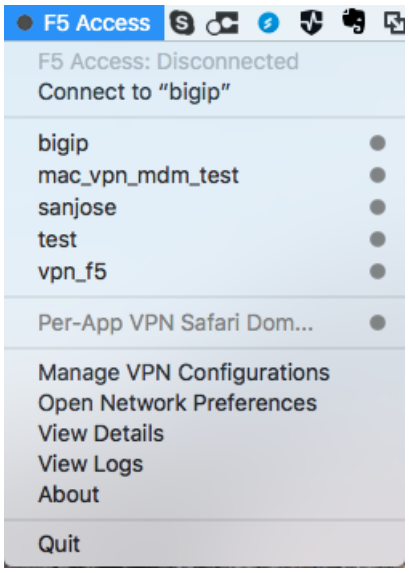
Web-based Authentication is supported in this version. In web authentication mode, the administrator can specify interactive Web-based multi-factor authentication in the access policy. Web authentication mode can be used to support an external logon page, SAML authentication, 2-factor logon with a one-time passcode, or other interactive methods. A user can specify Web logon mode when creating a configuration. All Web logon feature are supported.

Client certificate required mode

In this version, client certificate required mode is supported.

About the F5 Access for macOS container app

Container app interface



After F5 Access for macOS is installed, the container app is available from the macOS menu bar.

The following functions and status items can be viewed and accessed from the container app:

- **Connection status**: Shows the status of F5 Access, and the status of configured connections.
- **Connect to**: This menu item allows you to connect to a VPN from the list of VPN configurations. **Connect to** defaults to the most recently used configuration.
- **List of VPN configurations**: Shows the current configured VPN connections. The user can click a configuration to connect. Clicking another VPN configuration when connected causes the connection to switch VPN configurations.
- **Manage VPN Configurations**: Allows the user to add, edit, and remove VPN configurations. Note that configurations managed by a Mobile Device Manager (MDM) cannot be removed by the user.
- **Open Network Preferences**: Opens the network settings in the System Preferences app.
- **View Details**: Displays the connection details window.
- **View Logs**: Views the F5 Access logs. This can be useful for troubleshooting.
- **About F5 Access**: Shows information about the installed version of F5 Access
- **Quit**: Quits the container app. Note that this does not terminate the VPN connection.

Creating a VPN configuration from Container app

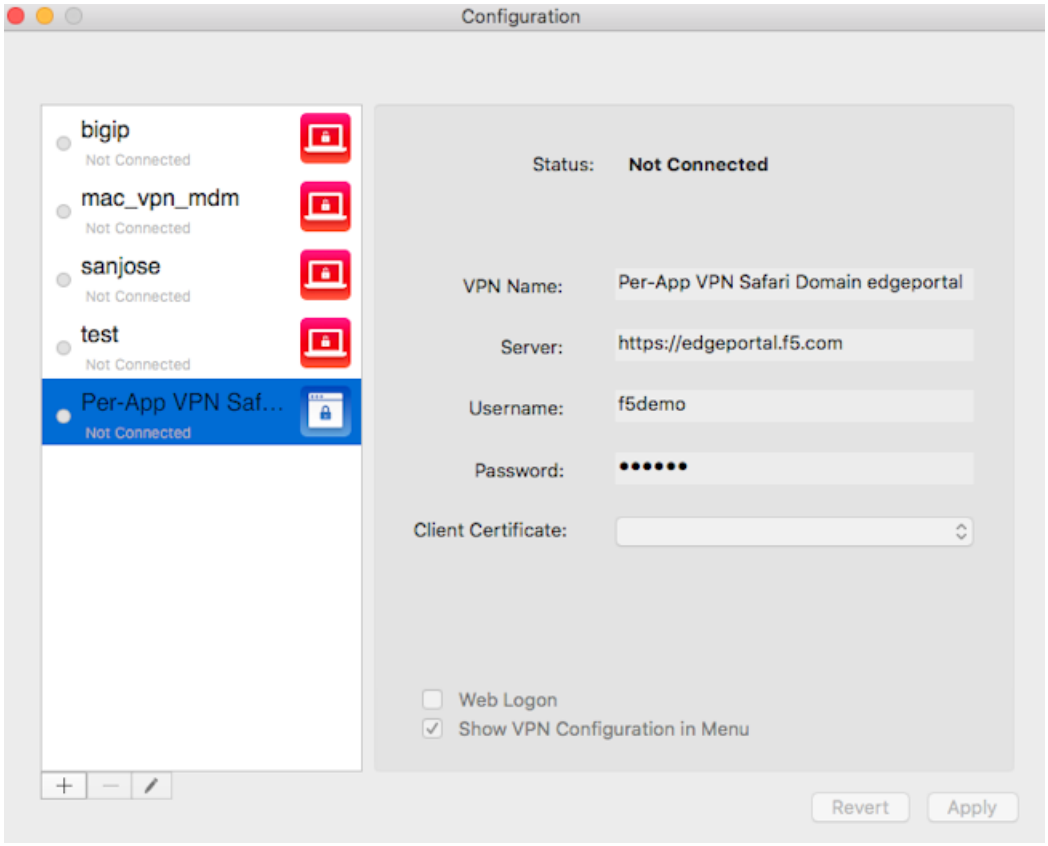
You create a configuration to establish a VPN connection to access network resources.

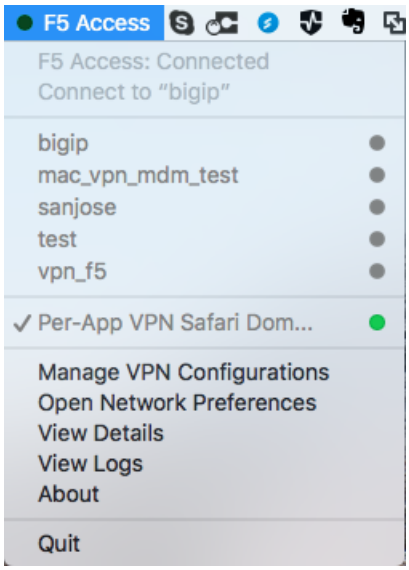
1. From the F5 Container App click **Manage VPN Configurations**.
2. Click + to add a configuration.
3. In the **VPN Name** field, type a VPN name.
4. In the **Server** field, type the server address.

5. In the **Username** field, type the username.
6. In the **Password** field, type the password.
7. In the **Client Certificate** field, select a common name for the client certificate.
8. To use a web page for logon, click **Web Logon**.
9. To show the VPN Configuration in the F5 Container App menu, click **Show VPN Configuration in Menu**.
10. Click **Apply**.

The VPN configuration is created. Start the VPN connection by selecting the configuration name from the F5 Container App menu.

Creating the Configuration in the Container App menu





Editing a VPN configuration from Container app

You can edit or delete a configuration from the Container app after you have created it.

1. From the F5 Container App click **Manage VPN Configurations**.
2. Click the name of a VPN configuration.
3. To edit the configuration, click the pencil icon.
4. To delete the configuration, click the minus icon.

Creating a VPN configuration from a plist file

You create a configuration to establish a VPN connection to access network resources.

Important: *You cannot edit or delete a VPN configuration created with a plist file from the VPN configurations dialog, though you can modify the username and password.*

Double-click a plist *.mobileconfig* file to install the VPN.

The VPN configuration is created. Start the VPN connection by selecting the configuration name from the F5 Container App menu.

Example plist mobileconfig VPN configuration file

This is a sample plist .mobileconfig file for VPN configuration.

VPN configuration with plist file example

The following example uses sample data only. For your own configuration, items like the `PayloadDisplayName`, `Payload UUID`, username, password and certificate information must be customized to your network and installation.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
    <key>PayloadContent</key>
    <array>
        <dict>
            <key>IPv4</key>
            <dict>
                <key>OverridePrimary</key>
                <integer>0</integer>
            </dict>
            <key>PayloadDescription</key>
          <string>Configures VPN settings, including authentication.</string>

            <key>PayloadDisplayName</key>
            <string>VPN (test_vpn_config)</string>
            <key>PayloadIdentifier</key>
            <string>com.f5.access.macos.vpn.profile</string>
            <key>PayloadOrganization</key>
            <string></string>
            <key>PayloadType</key>
            <string>com.apple.vpn.managed</string>
            <key>PayloadUUID</key>
            <string>3A0ED411-G45D-4551-AE35-650CE54B08D5</string>
            <key>PayloadVersion</key>
            <integer>1</integer>
            <key>Proxies</key>
            <dict/>
            <key>UserDefinedName</key>
            <string>test_vpn_config</string>
            <key>VPN</key>
            <dict>
                <key>AuthName</key>
                <string>username</string>
                <key>AuthPassword</key>
                <string>password</string>
                <key>AuthenticationMethod</key>
                <string>Password+Certificate</string>
                    <key>PayloadCertificateUUID</key>
                    <string>CF12345D-E819-4521-88DE-2AEB6E1DC3D8</string>
                <key>RemoteAddress</key>
                <string>https://selfip.example.com</string>
                <key>ProviderType</key>
                    <string>packet-tunnel</string>
                    <key>ProviderBundleIdentifier</key>
                    <string>com.f5.access.macos.PacketTunnel</string>
            </dict>
            <key>VPNSubType</key>
            <string>com.f5.access.macos</string>
            <key>VPNType</key>
            <string>VPN</string>
            <key>VendorConfig</key>
            <dict/>
        </dict>
        <dict>
                <key>Password</key>
                <string>123456</string>
                <key>PayloadCertificateFileName</key>
                <string>identity.p12</string>
                <key>PayloadContent</key>
```

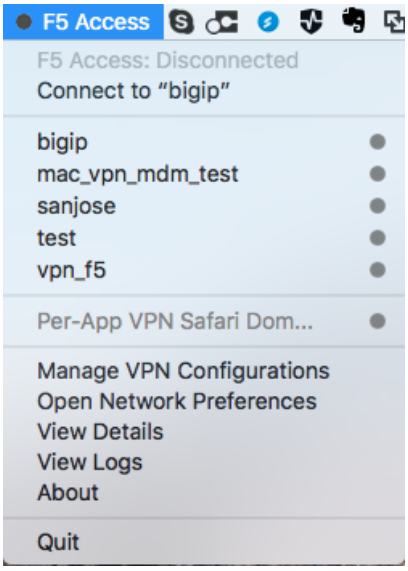
```
                <data>
MIIJCQIBAzCCCM8GCSqGSIb3DQEHAaCCCMAEggi8MIIIuDCCA28GCSqGSIb3DQEHBqCCA2AwggNcAgEA
MIIDVQYJKoZIhvcNAQcBMBwGCiqGSIb3DQEMAQYwDgQIzdOkMx7b/skCAggAgIIDKNjtUzTS2/diyoiU
ArGTs6vaAcb6PW7bjR/5gObmwV+NHT4BVqGVfm9L+F7zkhgtSx/gTVISOLphruYjSdpiqVN8IVcL6uVR
... (etc...)
            </data>
             <key>PayloadDescription</key>
             <string>Provides device authentication (certificate or
identity).</string>
             <key>PayloadDisplayName</key>
             <string>identity.p12</string>
             <key>PayloadIdentifier</key>
             <string>com.f5.access.macos.vpn.credential</string>
             <key>PayloadOrganization</key>
             <string/>
             <key>PayloadType</key>
             <string>com.apple.security.pkcs12</string>
             <key>PayloadUUID</key>
             <string>C9BF4927-E819-4521-88DE-2AEB6E1DC3D8</string>
             <key>PayloadVersion</key>
             <integer>1</integer>
            </dict>
    </array>
    <key>PayloadDescription</key>
    <string>f5 mac tunnel test</string>
    <key>PayloadDisplayName</key>
    <string>mac_vpn_mdm_profile</string>
    <key>PayloadIdentifier</key>
    <string>com.f5.access.macos.vpn.profile</string>
    <key>PayloadOrganization</key>
    <string></string>
    <key>PayloadRemovalDisallowed</key>
    <false/>
    <key>PayloadType</key>
    <string>Configuration</string>
    <key>PayloadUUID</key>
    <string>A6F83919-B570-41FE-A84F-52DAC24838D8</string>
    <key>PayloadVersion</key>
    <integer>1</integer>
</dict>
</plist>
```

Starting a connection manually

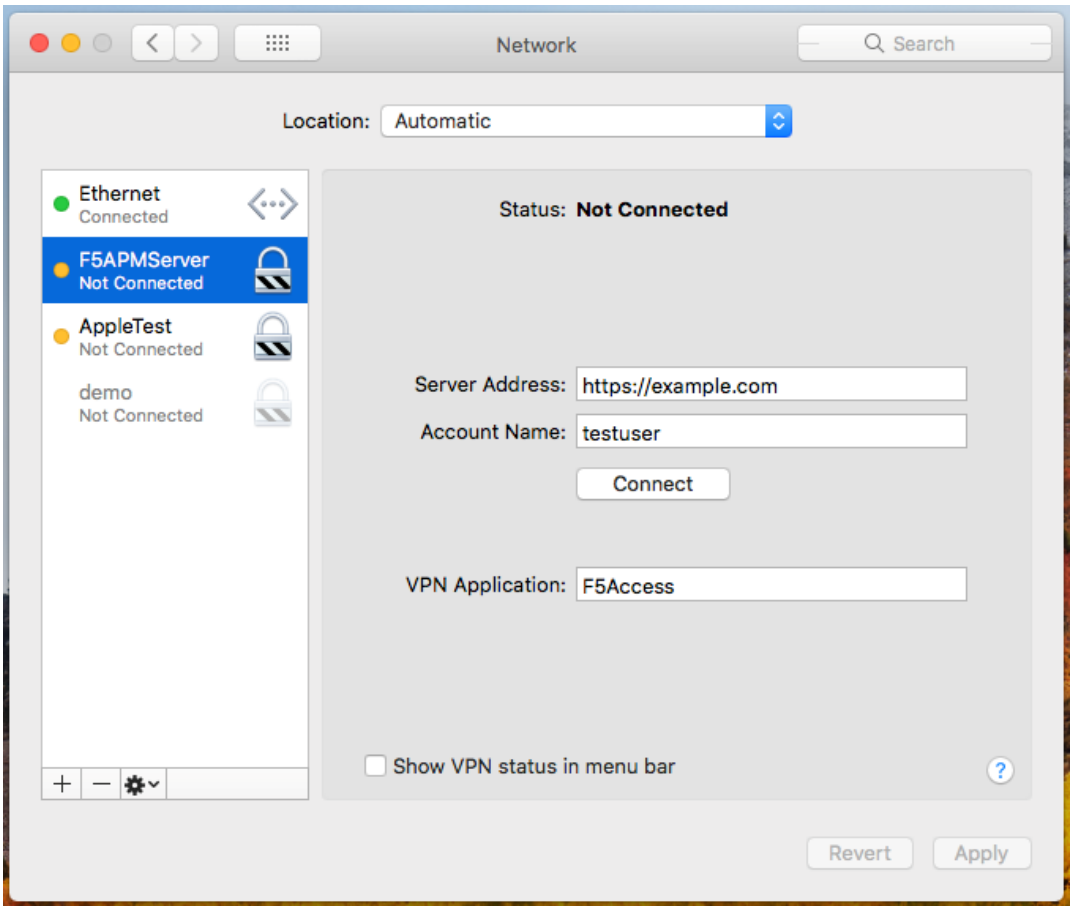
Starting a connection on F5 Access for macOS requires a configured BIG-IP Network Access access policy to which you can connect. All configurations created from the Container app are also available in the **System Preference > Network** panel.

You start a connection to access network resources.

1. Log in to the macOS device and launch the F5 Access application from the Finder or the Launch Pad.
2. Start a connection by selecting an exisiting connection from the list.



3. You can also start a connection from the **System Preference > Network** panel.



Configuring Access Policy Manager for F5 Access

What does F5 Access do for macOS devices?

F5 Access for macOS provides full network access through BIG-IP® Access Policy Manager®. With network access, users can run applications such as RDP, SSH, Citrix, VMware View, and other enterprise applications on their macOS devices.

F5 Access features include:

- User name and password, and client certificate support
- Support for DNS address space for split-tunneling configurations
- Landing URI support
- Logging support to report issues
- Support client certificate for DTLS tunnels and SSL tunnels
- Per-app VPN support
- Password caching support

About supported authentication types

F5 Access for macOS provides these authentication types:

Authentication type	Connection Type
Client certificate	• User-initiated connections, in native mode or Web Logon mode • Device-wide VPN On-Demand, in native mode or Web Logon mode • Per-App VPN connections, in native mode only Per-App VPN does not support Web Logon mode.
Client certificate + username and password	Runtime prompts (login dialogs, and other user input prompts) are allowed for: • User-initiated connections, in native mode or Web Logon mode • Device-wide VPN On-Demand connections, in native mode or Web Logon mode • Per-App VPN connections, in native mode only Per-App VPN does not support Web Logon mode.
Username and password	Runtime prompts (login dialogs, and other user input prompts) are allowed for: • User-initiated connections, in native mode or Web Logon mode • Device-wide VPN On-Demand connections, in native mode or Web Logon mode • Per-App VPN connections, in native mode only Per-App VPN does not support Web Logon mode.

About establishing VPN connections

The F5 Access application (app) for macOS devices provides users with two options to establish a VPN tunnel connection. A user can start a tunnel connection explicitly with the F5 Access application, or implicitly through the VPN On-Demand functionality.

For example, a connection can be configured to automatically trigger whenever a certain domain or host name pattern is matched.

About pre-logon checks supported for macOS devices

For macOS devices, Access Policy Manager® can use only the following preconfigured pre-logon checks:

- Client Type - result is F5 Access
- Client OS - result is MacOS

Other session variables can be checked using custom expressions. See the list of session variables for macOS for more information.

Setting up network access

You can force traffic through a tunnel on F5 Access.

Note: *Although you disable* ***Allow local subnet access*** *while enabling* ***Force all traffic through tunnel****, the client still permits local subnet traffic to travel outside of the tunnel. This is a limitation of macOSand not of F5 Access.*

---

1. On the Main tab, click **Access Policy** > **Network Access** > **Network Access List**.
   The Network Access List screen opens.
2. Click the name to select a network access resource on the Resource List.
   The Network Access editing screen opens.
3. To configure the network settings for the network access resource, click **Network Settings** on the menu bar.
4. To optionally force all traffic through the tunnel, next to **Traffic Options**, enable **Force all traffic through tunnel**.

   If you enable **Use split tunneling for traffic**, you must also specify either a DNS suffix or DNS Address Space pattern to use the VPN DNS servers. If the "DNS Suffix" and "DNS Address Space" fields are both left blank, then F5 Access does not use the VPN DNS servers and sends all DNS traffic to public DNS servers.
5. To allow local subnet traffic to bypass the tunnel, select the **Enable** check box for **Allow Local Subnet**. This traffic bypasses the tunnel.
6. Click **Update**.

Configuring the connectivity profile for macOS

You can configure password caching and enforce native or web logon mode by configuring the connectivity profile.

1. On the Main tab, click **Access** > **Connectivity / VPN** > **Connectivity** > **Profiles**.
   The Connectivity Profiles screen opens.
2. Click the name of the Connectivity profile that you use with F5 Access for macOS, and click **Edit Profile**.
3. Click the **F5 Access for macOS** item to configure F5 Access for macOS settings.
4. To allow password caching on the macOS client, click **Allow Password Caching**. From the Save Password Method list, select **disk** or **memory**.

   If you select **disk**, an encrypted password is saved on disk with no expiration time. If you select **memory**, an encrypted password is cached on the device for the time specified in the **Password Cache Expiration (minutes)** field. The default value is **240** minutes (4 hours).
5. To enforce the logon mode, click **Enforce Logon Mode**. Select **native** or **web** for the logon mode.

   If **Enforce Logon Mode** is enabled in the Connectivity Profile, the user cannot change the Web Logon option.
6. Click **OK**.

Prerequisites for configuring F5 Access

Before configuring F5 Access for macOS devices, you must complete the following requirements:

- Set up BIG-IP® Access Policy Manager®.
- Run the Network Access Setup Wizard.

Additional information about network access and connectivity profiles can be found in the *BIG-IP® Access Policy Manager®: Network Access Configuration* guide.

Access Policy Manager configuration for F5 Access for macOS devices

To configure F5 Access for macOS device support on BIG-IP® Access Policy Manager®, use the following configuration steps:

- Run the Network Access Setup Wizard.
- Optionally, set up `SSO` and `ACLs` for your network access. Refer to the *BIG-IP® Access Policy Manager® Configuration Guide* on the AskF5™ Knowledge Base for instructions.
- Customize an access policy to support F5 Access.

Running the Network Access Setup wizard

Configure Access Policy Manager® to provide users with full network access from their devices using the Network Access Setup wizard for remote access.

1. On the Main tab, click **Wizards** > **Device Wizards**.
   The Device Wizards screen opens.
2. For Access Policy Manager Configuration, select **Network Access Setup Wizard for Remote Access**, and then click **Next**.
3. Click **Finished**.

You now have network access resource that supports F5 Access for mobile devices.

Overview: Access Policies for F5 Access

About access policy branches for F5 Access

You can configure separate access policy branches for F5 Access.

F5 Access does not support client-side checks; however, you can configure an access policy that provides network access for macOS clients by using any of these methods:

- Create an access policy using **Client-Side Capability**. This provides a branch for clients that do not support client-side checks. Assign authentication and a network access resource to this branch.
- Use an existing access policy with client-side checks. The macOS client will fail to the fallback branch of the first client-side check, if the logon mode is native. If the logon mode is Web Logon, user interaction is required to proceed with the fallback branch. Assign authentication and a network access resource to the fallback branch.
- Add a **Client OS** Access Policy item, and assign authentication and resources to the **macOS** branch.

F5 Access for macOS is detected with the following access policy items:

Access policy item	Value
Client Type	F5 Access
Client OS	MacOS

Configuring an access policy for F5 Access for macOS

Configure an access policy to identify and allow access to macOS devices.

1. On the Main tab, click **Access** > **Profiles / Policies**.
   The Access Profiles List screen opens.
2. Click **Create**.
   The New Profile screen opens.
3. In the **Name** field, type a name for the access profile.

   Note: *An access profile name must be unique among all access profile and any per-request policy names.*

4. From the **Profile Type** list, select **SSL-VPN**.
   Additional settings display.
5. From the **Profile Scope** list, retain the default value or select another.
   - **Profile**: Gives a user access only to resources that are behind the same access profile. This is the default value.
   - **Virtual Server**: Gives a user access only to resources that are behind the same virtual server.
   - **Global**: Gives a user access to resources behind any access profile that has global scope.
6. In the Language Settings area, add and remove accepted languages, and set the default language.

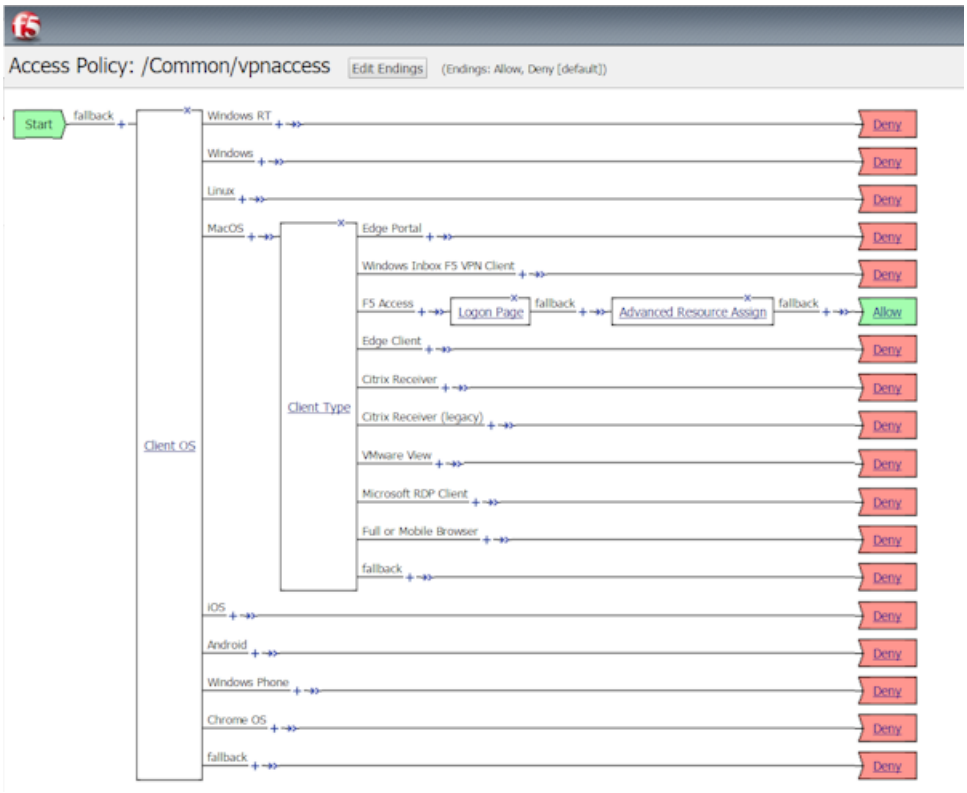
A browser uses the highest priority accepted language. If no browser language matches the accepted languages list, the browser uses the default language.

7. Click **Finished**.
8. Click the name of the access profile for which you want to edit the access policy.
   The properties screen opens for the profile you want to edit.
9. Click the **Access Policy** tab.
10. In the General Properties area, click the **Edit Access Policy for Profile** ***profile_name*** link.
    The visual policy editor opens the access policy in a separate screen.
11. Click **Add Item**.
    The screen is not active while the visual policy editor creates the action. The screen closes and a Properties screen displays.
12. Click the Endpoint Security (Server-Side) tab, and select Client OS.
13. Click **Add Item**.
    The screen is not active while the visual policy editor creates the action. The screen closes and a Properties screen displays.
14. Click **Save**.
15. On the MacOS branch, click **Add Item**.
16. Click the Endpoint Security (Server-Side) tab, and select Client Type.
17. Click **Save**.
18. On the F5 Access branch, add the authentication and resource actions you require. For example, add a Logon Page, Client Certificate, and Resource Assign actions.
19. When you have finished configuring the access policy, click **Apply Access Policy**.

Example of basic access policy that supports F5 Access

You can configure an access policy branch to direct macOS device users to F5 Access, and direct non-F5 Access device users to a fallback branch.

This example displays a simple access policy.


Access Policy: /Common/vpnaccess
Edit Endings
(Endings: Allow, Deny [default])
Start
fallback
Client OS
Windows RT
Windows
Linux
MacOS
Client Type
Edge Portal
Windows Inbox F5 VPN Client
F5 Access
Logon Page
fallback
Advanced Resource Assign
fallback
Allow
Edge Client
Citrix Receiver
Citrix Receiver (legacy)
VMware View
Microsoft RDP Client
Full or Mobile Browser
fallback
iOS
Android
Windows Phone
Chrome OS
fallback
Deny

Configuring Per-App VPN with APM and F5 Access

What is per-app VPN?

Apple's VPN framework supports layer-3 tunneling for TCP and UDP connections. Apps can be configured to automatically connect to a VPN when they are started. Safari can be configured for per-app VPN with a configuration profile and without an MDM, and on a per-URL basis.

Note: *An access policy for Per-App VPN on macOS is similar to a device-wide VPN access policy, except that items that require Web Logon, such as multi-factor authentication, are not supported.*

A per-app VPN configuration requires two configuration components.

- A device under MDM management or a configuration profile installed manually. For more information, see *macOS Sierra: Use configuration profiles*.
- F5 Access for macOS installed on the device.

About deploying MDM apps over VPNs

The per-app VPN framework allows the administrator to limit VPN access to explicit apps only. Specifically, it allows applications to use one F5 Access configuration (or VPN connection).

Important: *If the F5 Access configuration is not connected when the app starts, all traffic from the app is blocked.*

In practice, some applications may be associated with one F5 Access configuration, and other applications may be associated with other F5 Access configurations.

Important: *Once an app is associated with an F5 Access configuration by the MDM, it must use that VPN only.*

In this example, only App 1 or App 2 can be active at one time.

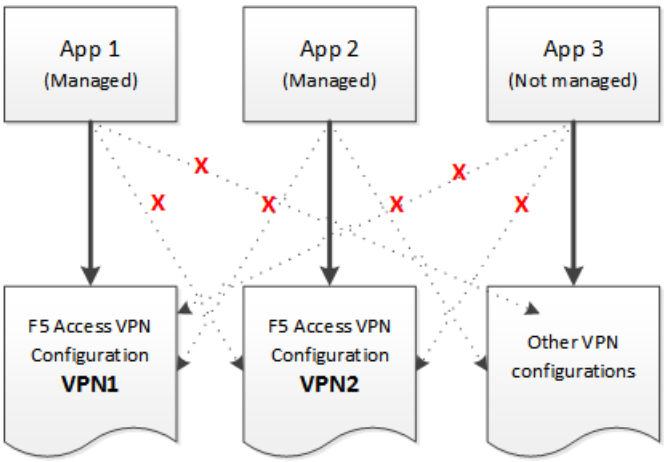


Figure 1: Apps associated with different VPN configurations

Note: *On macos, you can only activate one device-wide or Per-App VPN configuration at a time.*

About access policies for per-app VPN

For per-app VPN, an access policy requires a specific configuration. The per-app VPN process does allow prompts or requests for information (logon and password) during logon. However, Web Logon is not supported.

Creating an access profile

You create an access profile to provide the secured connection between the per-app VPN and the virtual server.

1. On the Main tab, click **Access Policy** > **Access Profiles**.
   The Access Profiles List screen opens.
2. Click **Create**.
   The New Profile screen opens.
3. In the **Name** field, type a name for the access profile.
4. In the Language Settings area, add and remove accepted languages, and set the default language.

   A browser uses the highest priority accepted language. If no browser language matches the accepted languages list, the browser uses the default language.
5. Click **Finished**.

The access profile appears in the Access Profiles List.

Adding a client certificate check to the access policy

A client certificate check or on-demand cert auth check allows you to authenticate the device to the access policy.

1. Click **Access** > **Profiles / Policies** > **Access Profiles (Per-Session Policies)**.
2. In the Per-Session Policy column, click the **Edit** link for the access profile you want to configure to launch the visual policy editor.
   The visual policy editor opens the access policy in a separate screen.
3. Click the **(+)** sign anywhere in the access policy to add a new action item.
   An Add Item screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. Click **Add Item**.
   The screen is not active while the visual policy editor creates the action. The screen closes and a Properties screen displays.
5. Click the **Authentication** tab.
6. Select the **Client Cert Inspection** item or **On-Demand Cert Auth** item, and click **Add Item**.
7. Click **Apply Access Policy** to save your configuration.
8. The properties screen opens. Click **Save**.
9. On the **Successful** branch following the Client Cert Inspection or On-Demand Cert Auth item, click the Deny ending.
10. Change the Deny ending to Allow, and click **Save**.
11. Click **Apply Access Policy** to save your configuration.

The access profile appears in the Access Profiles List.

Configure the virtual server to include this access policy, and make sure the Client SSL profile is enabled on the server.

Managing Devices for F5 Access

About managing devices

With an MDM, you manage devices by enrolling them. Refer to your MDM documentation to enroll devices. With this release, your MDM vendor may not include built-in support. We provide general guidance for your MDM configuration, if it supports custom configurations.

Important: *A user must enroll the device with the MDM in order for you to manage the device. However, you can deploy VPN configurations to the devices that aren't under management. F5 Access must be installed on the device to deploy configurations. F5 Access can be installed either by the user, or deployed with the MDM solution.*

Creating a configuration profile for the managed device

A configuration profile enables the per-app VPN feature on a managed device, and specifies which apps use the VPN.

Create a configuration profile for the device.

Configuration profiles are described at the *Apple Configuration Profile Reference*.

Configure Access Policy Manager ®to provide the necessary support for per-app VPN features.

Device identification configuration profile settings

These are settings for identifying devices in an MDM profile.

Device identification settings

Hardware manufacturers have phased out support for many methods of device identification, including UDID, wireless MAC, and others. To identify devices, you can use the device IDs assigned by the MDM.

Table 1: Device identification commands

Key	Type	Description
MdmAssignedId	String	The internal device ID assigned to the device by the MDM.
MdmInstanceId	String	An arbitrary string that identifies particular MDM instance.
MdmDeviceUniqueId	String	An assigned ID for the device.
MdmDeviceWifiMacAddress	String	The wireless MAC address of the device.
MdmDeviceSerialNumber	String	An assigned serial number for the device.

Device ID example for macOS

In this example, the commands are deployed in the VendorConfig document.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
...
    <key>VendorConfig</key>
    <dict>

       <key>MdmAssignedId</key>
       <string>MDM assigned ID here</string>
       <key>MdmInstanceId</key>
       <string>some MDM instance ID here</string>
       <key>MdmDeviceUniqueId</key>
       <string>device macOS UDID here</string>
       <key>MdmDeviceWifiMacAddress</key>
       <string>device wifi mac address here</string>
       <key>MdmDeviceSerialNumber</key>
       <string>device serial number here</string>
    </dict>
...
```

Always On VPN setting

This setting configures Always On mode in an MDM profile.

Always On configuration

In the MDM configuration profile, you can enable Always On VPN by setting an On Demand rule with the key URLStringProbe. This allows the On Demand VPN to start as soon as the "probed" URL is contacted.

The code is used as follows.

```
<key>OnDemandRules</key>
   <array>
    <dict>
     <key>Action</key>
     <string>Connect</string>
     <key>URLStringProbe</key>
     <array>
      <string>https://on.example.com</string>
     </array>
    </dict>
   </array>
```

Example Always On VPN configuration profile

Includes a sample configuration profile for an Always On VPN profile.

Always On VPN configuration example profile

The following example uses sample data only. For your own configuration, items like the `PayloadDisplayName`, `PayloadUUID`, `UserDefinedName`, and the user name, password and certificate information must be customized to your network and installation.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
```

```
<plist version="1.0">
<dict>
 <key>PayloadContent</key>
 <array>
  <dict>
   <key>IPv4</key>
   <dict>
    <key>OverridePrimary</key>
    <integer>0</integer>
   </dict>
   <key>PayloadDescription</key>
   <string>Configures VPN settings, including authentication.</string>
   <key>PayloadDisplayName</key>
   <string>Always on VPN (URL probe)</string>
   <key>PayloadIdentifier</key>
   <string>com.f5.access.macos.vpn.profile</string>
   <key>PayloadOrganization</key>
   <string></string>
   <key>PayloadType</key>
   <string>com.apple.vpn.managed</string>
   <key>PayloadUUID</key>
   <string>3A0ED411-E55C-4551-AE35-950CE54B08D5</string>
   <key>PayloadVersion</key>
   <integer>1</integer>
   <key>Proxies</key>
   <dict/>
   <key>UserDefinedName</key>
   <string>OnDemandAlwaysOnConfig</string>
   <key>VPN</key>
   <dict>
    <key>AuthName</key>
    <string>username</string>
    <key>AuthPassword</key>
    <string>password</string>
    <key>AuthenticationMethod</key>
    <string>Password</string>
    <key>RemoteAddress</key>
    <string>https://vpn.example.com</string>
    <key>ProviderType</key>
             <string>packet-tunnel</string>
             <key>ProviderBundleIdentifier</key>
             <string>com.f5.access.macos.PacketTunnel</string>
   </dict>
   <key>VPNSubType</key>
   <string>com.f5.access.macos</string>
   <key>VPNType</key>
   <string>VPN</string>
   <key>VendorConfig</key>
   <dict/>
   <key>OnDemandEnabled</key>
   <integer>1</integer>
   <key>OnDemandRules</key>
   <array>
    <dict>
     <key>Action</key>
     <string>Connect</string>
     <key>URLStringProbe</key>
     <array>
      <string>https://vpn.example.com</string>
     </array>
    </dict>
   </array>
  </dict>
 </array>
 <key>PayloadDescription</key>
 <string>F5Access test</string>
 <key>PayloadDisplayName</key>
 <string>OnDemandRuleURLProbeTest</string>
 <key>PayloadIdentifier</key>
```

```
 <string>com.f5.access.macos.vpn.profile</string>
 <key>PayloadOrganization</key>
 <string></string>
 <key>PayloadRemovalDisallowed</key>
 <false/>
 <key>PayloadType</key>
 <string>Configuration</string>
 <key>PayloadUUID</key>
 <string>A6F83919-B570-41FE-A84F-52DAC24838E8</string>
 <key>PayloadVersion</key>
 <integer>1</integer>
</dict>
</plist>
```

Device-wide VPN configuration profile settings

Settings for the device-wide VPN profiles in an MDM configuration.

Device-wide VPN settings

Configure a device-wide VPN by specifyng the VPN payload. For the *PayloadType* value, specify `com.apple.vpn.managed`. F5 Access 2.0 VPN configurations must define the following keys:

Table 2: System-Wide VPN specific keys

Key	Type	Description
PayloadType	String	`com.apple.vpn.managed`
VPNType	String	`VPN`
VPNSubType	String	`com.f5.access.macos`
ProviderBundleIdentifier	String	Optional key: `com.f5.access.macos.PacketTunnel`
OnDemandEnabled	Int	Optional key: `1` if the VPN connection should be brought up on demand, or else `0`.
OnDemandRules	Array of Dictionaries	Optional key. Determines when and how an on-demand VPN should be used. See *On Demand Rules Dictionary Keys* for details.

Example device-wide VPN configuration profile

Includes a sample configuration profile for the device-wide VPN configuration profile.

Device-wide VPN configuration example profile

The following example uses sample data only. For your own configuration, items like the `PayloadDisplayName`, `PayloadUUID`, `UserDefinedName`, and the user name, password and certificate information must be customized to your network and installation.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
 <key>PayloadContent</key>
 <array>
  <dict>
```

```
   <key>IPv4</key>
   <dict>
    <key>OverridePrimary</key>
    <integer>0</integer>
   </dict>
   <key>PayloadDescription</key>
   <string>Configures VPN settings, including authentication.</string>
   <key>PayloadDisplayName</key>
   <string>VPN (Client Certificate)</string>
   <key>PayloadIdentifier</key>
   <string>com.f5.access.macos.vpn.profile</string>
   <key>PayloadOrganization</key>
   <string></string>
   <key>PayloadType</key>
   <string>com.apple.vpn.managed</string>
   <key>PayloadUUID</key>
   <string>3A0ED411-E55C-4551-AE35-650CE54B08D5</string>
   <key>PayloadVersion</key>
   <integer>1</integer>
   <key>Proxies</key>
   <dict/>
   <key>UserDefinedName</key>
   <string>vpn_test</string>
   <key>VPN</key>
   <dict>
    <key>AuthName</key>
    <string>user_name</string>
    <key>AuthPassword</key>
    <string>user_password</string>
    <key>AuthenticationMethod</key>
    <string>Password+Certificate</string>
             <key>PayloadCertificateUUID</key>
             <string>C9BF4927-E819-4521-88DE-2AEB6E1DC3D8</string>
    <key>RemoteAddress</key>
    <string>https://test.lab.example.com</string>
    <key>ProviderType</key>
             <string>packet-tunnel</string>
             <key>ProviderBundleIdentifier</key>
             <string>com.f5.access.macos.PacketTunnel</string>
   </dict>
   <key>VPNSubType</key>
   <string>com.f5.access.macos</string>
   <key>VPNType</key>
   <string>VPN</string>
   <key>VendorConfig</key>
   <dict/>
  </dict>
  <dict>
          <key>Password</key>
          <string>123456</string>
          <key>PayloadCertificateFileName</key>
          <string>identity.p12</string>
          <key>PayloadContent</key>
          <data>
MIIJCQIBAzCCCM8GCSqGSIb3DQEHAaCCCMAEggi8MIIIuDCCA28GCSqGSIb3DQEHBqCCA2AwggNcAgEA
MIIDVQYJKoZIhvcNAQcBMBwGCiqGSIb3DQEMAQYwDgQIzdOkMx7b/skCAggAgIIDKNjtUzTS2/diyoiU
...
1Ez4mnbrWzElMCMGCSqGSIb3DQEJFTEWBBQAFBOqYFJlbkBoqPfCMK5F1BXODDAxMCEwCQYFKw4DAhoF
AAQUqF+54GDMxB3FcOmVKmAOIMKzxl8ECAKbcibSFUHZAgIIAA==
          </data>
         <key>PayloadDescription</key>
         <string>Provides device authentication (certificate or
identity).</string>
         <key>PayloadDisplayName</key>
         <string>identity.p12</string>
         <key>PayloadIdentifier</key>
         <string>com.f5.access.macos.vpn.credential</string>
         <key>PayloadOrganization</key>
         <string/>
```

```
        <key>PayloadType</key>
        <string>com.apple.security.pkcs12</string>
        <key>PayloadUUID</key>
        <string>C9BF4927-E819-4521-88DE-2AEB6E1DC3D8</string>
        <key>PayloadVersion</key>
        <integer>1</integer>
         </dict>
 </array>
 <key>PayloadDescription</key>
 <string>f5 mac tunnel test</string>
 <key>PayloadDisplayName</key>
 <string>mac_vpn_mdm_profile</string>
 <key>PayloadIdentifier</key>
 <string>com.f5.access.macos.vpn.profile</string>
 <key>PayloadOrganization</key>
 <string></string>
 <key>PayloadRemovalDisallowed</key>
 <false/>
 <key>PayloadType</key>
 <string>Configuration</string>
 <key>PayloadUUID</key>
 <string>A6F83919-B570-41FE-A84F-52DAC24838D8</string>
 <key>PayloadVersion</key>
 <integer>1</integer>
</dict>
</plist>
```

Per-App VPN configuration profile settings

Settings for the per-app VPN profile in an MDM.

Per-App VPN settings

The per-app VPN payload supports all of the keys described in the *Apple Configuration Profile Reference*. These keys, specific to the per-app VPN payload, are described in that reference as well.

Table 3: Per-App VPN keys

Key	Type	Description
PayloadType	String	`com.apple.vpn.managed.applayer`
VPNType	String	`VPN`
ProviderType	String	`packet-tunnel`
VPNSubType	String	`com.f5.access.macos`
ProviderBundleIdentifier	String	Optional key: `com.f5.access.macos.PacketTunnel`
VPNUUID	String	A globally-unique identifier for this VPN configuration. This identifier is used to configure apps so that they use the per-app VPN service for all of their network communication.
OnDemandMatchAppEnabled (optional)	Boolean	If `true`, the per-app VPN connection starts automatically when apps linked to this per-app VPN service initiate network communication. If `false`, the per-app VPN connection will not start.

Key	Type	Description
		If this key is not present, the value of the OnDemandEnabled key is used to determine the status of per-app VPN On Demand.
SafariDomains (optional)	Array	This key is a special case of App-to-Per App VPN Mapping. It sets up the app mapping for Safari with a specific identifier and a designated requirement. The array contains strings, each of which is a domain that triggers a VPN connection in Safari. Do not specify a full URI; rule matching works only with the domain name. The rule matching behavior is as follows: • Before being matched against a host, all leading and trailing dots are stripped from the domain string. For example, if the domain string is `.com` the domain string used to match is `com`. • Each label in the domain string must match an entire label in the host string. For example, a domain of `example.com` matches "`www.example.com`", but not `old.badexample.com`. • Domain strings with only one label must match the entire host string. For example, a domain of `com` matches `com`, not `www.example.com`.

Example per-app VPN configuration profile

Includes a sample configuration profile for the per-app VPN configuration profile.

Per-App VPN configuration example profile

The following example uses sample data only. For your own configuration, items like the `PayloadDisplayName`, `PayloadUUID`, `UserDefinedName`, and the user name, password and certificate information must be customized to your network and installation.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
  <dict>
    <key>PayloadContent</key>
    <array>
      <dict>
        <key>IPv4</key>
        <dict>
          <key>OverridePrimary</key>
          <integer>0</integer>
        </dict>
        <key>PayloadDescription</key>
        <string>Configures VPN settings, including authentication.</string>
        <key>PayloadDisplayName</key>
        <string>VPN (Per-App VPN Test)</string>
        <key>PayloadIdentifier</key>
        <string>com.f5.mdm.perapp.vpn.vpn</string>
        <key>PayloadOrganization</key>
        <string/>
        <key>PayloadType</key>
```

```
        <string>com.apple.vpn.managed.applayer</string>
        <key>PayloadUUID</key>
        <string>5A015006-D559-4C5C-B197-737CF4DCFA96</string>
        <key>PayloadVersion</key>
        <integer>1</integer>
        <key>Proxies</key>
        <dict/>
        <key>UserDefinedName</key>
        <string>Per-App VPN Test</string>
        <key>VPN</key>
        <dict>
          <key>AuthName</key>
          <string>test</string>
          <key>AuthPassword</key>
          <string>test</string>
          <key>AuthenticationMethod</key>
          <string>Certificate</string>
          <key>PayloadCertificateUUID</key>
          <string>C9BF4927-E819-4521-88DE-2AEB6E1DC3D8</string>
          <key>RemoteAddress</key>
          <string>https://portal.example.com</string>
          <key>OnDemandMatchAppEnabled</key>
          <true/>
          <key>ProviderType</key>
          <string>packet-tunnel</string>
          <key>ProviderBundleIdentifier</key>
          <string>com.f5.access.macos.PacketTunnel</string>
        </dict>
        <key>VPNSubType</key>
        <string>com.f5.access.macos</string>
        <key>VPNType</key>
        <string>VPN</string>
        <key>VendorConfig</key>
        <dict/>
        <key>SafariDomains</key>
        <array>
          <string>example.com</string>
          <string>main.example.com</string>
        </array>
        <key>VPNUUID</key>
        <string>FEC8510C-3F8A-4C51-8EFC-21A21D485C3C</string>
      </dict>
      <dict>
        <key>Password</key>
        <string>123456</string>
        <key>PayloadCertificateFileName</key>
        <string>identity.p12</string>
        <key>PayloadContent</key>
        <data>
            MIIL2QIBAzCCC58GCSqGSIb3DQEHAaCCC5AEgguMMIILiDCCBj8G
            CSqGSIb3DQEHBqCCBjAwggYsAgEAMIIGJQYJKoZIhvcNAQcBMBwG
            ...
            BBQAFBOqYFJlbkBoqPfCMK5F1BXODDAxMCEwCQYFKw4DAhoFAAQU
            hxd6YPi7JKB/24dSls9gKO/DHVoECHap2RUyKvQTAgIIAA==
            </data>
        <key>PayloadDescription</key>
        <string>Provides device authentication (certificate or
identity).</string>
        <key>PayloadDisplayName</key>
        <string>identity.p12</string>
        <key>PayloadIdentifier</key>
        <string>com.f5.mdm.perapp.vpn.credential</string>
        <key>PayloadOrganization</key>
        <string/>
        <key>PayloadType</key>
        <string>com.apple.security.pkcs12</string>
        <key>PayloadUUID</key>
        <string>C9BF4927-E819-4521-88DE-2AEB6E1DC3D8</string>
        <key>PayloadVersion</key>
```

```
        <integer>1</integer>
      </dict>
    </array>
    <key>PayloadDescription</key>
    <string>PerApp VPN Payload Test</string>
    <key>PayloadDisplayName</key>
    <string>MDM - Per-App VPN</string>
    <key>PayloadIdentifier</key>
    <string>com.f5.mdm.perapp.vpn</string>
    <key>PayloadOrganization</key>
    <string/>
    <key>PayloadRemovalDisallowed</key>
    <false/>
    <key>PayloadType</key>
    <string>Configuration</string>
    <key>PayloadUUID</key>
    <string>06A850CC-BC81-43FB-AA16-42BE472D2421</string>
    <key>PayloadVersion</key>
    <integer>1</integer>
  </dict>
</plist>
```

App to Per-App VPN mapping

Settings for mapping an app to Per-App VPN.

App to Per-App VPN mapping

If you want applications to use Per-App VPN service for all of their network communication, add a payload dictionary with the following keys to the configuration profile.

Table 4: App to Per-App VPN payload dictionary keys

Key	Type	Value
PayloadType	String	`com.apple.vpn.managed.appmapping`
AppLayerVPNMapping	Array of dictionaries	An array of mapping dictionaries.

Table 5: AppLayerVPNMapping dictionary entry array keys

Key	Type	Value

Identifier	String	The app's bundle ID. For example, for Google Chrome:
		`Identifier: com.google.Chrome`
		`Identifier: org.mozilla.firefox`

VPNUUID	String	The VPNUUID of the Per-App VPN defined in a Per-App VPN payload
DesignatedRequirement	String	The code signature designated requirement of the app that will use the Per-App VPN.
SigningIdentifier	String	The code signature signing identifier of the app that will use the Per-App VPN.

Example app to Per-App VPN mapping profile

Includes a sample configuration profile for app to per-app VPN mapping.

App to Per-App VPN configuration example profile

The following example uses sample data only. For your own configuration, items like the `PayloadDisplayName`, `PayloadUUID`, `UserDefinedName`, and the user name, password and certificate information must be customized to your network and installation.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN"
"http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
   <dict>
   <key>PayloadContent</key>
   <array>
      <dict>
         <key>PayloadDescription</key>
         <string>Configures VPN app mapping settings</string>
         <key>PayloadDisplayName</key>
         <string>VPN (Per-App VPN TCP App Mapping)</string>
         <key>PayloadIdentifier</key>
         <string>com.f5.access.macos.perapp.vpn.appmapping</string>
         <key>PayloadOrganization</key>
         <string/>
         <key>PayloadType</key>
         <string>com.apple.vpn.managed.appmapping</string>
         <key>PayloadUUID</key>
         <string>6B015006-D559-4C5C-B197-737CF4DCFA96</string>
         <key>PayloadVersion</key>
         <integer>1</integer>
         <key>AppLayerVPNMapping</key>
         <array>
            <dict>
               <key>Identifier</key>
               <string>com.google.Chrome</string>
               <key>VPNUUID</key>
               <string>6A015006-D559-4C5C-B197-737CF4DCFA96</string>
               <key>SigningIdentifier</key>
               <string>com.google.Chrome</string>
               <key>DesignatedRequirement</key>
               <string>(identifier "com.google.Chrome" or identifier
"com.google.Chrome.beta" or identifier "com.google.Chrome.dev" or identifier
 "com.google.Chrome.canary") and (certificate leaf =
H"85cee8254216185620ddc8851c7a9fc4dfe120ef" or certificate leaf =
H"c9a99324ca3fcb23dbcc36bd5fd4f9753305130a")</string>
            </dict>
            <dict>
               <key>Identifier</key>
               <string>org.mozilla.firefox</string>
               <key>VPNUUID</key>
               <string>6A015006-D559-4C5C-B197-737CF4DCFA96</string>
               <key>SigningIdentifier</key>
               <string>org.mozilla.firefox</string>
               <key>DesignatedRequirement</key>
               <string>anchor apple generic and certificate
leaf[field.1.2.840.113635.100.6.1.9] /* exists */ or anchor apple generic and
 certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate
leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate
leaf[subject.OU] = "43AQ936H96"</string>
               </dict>
         </array>
      </dict>
      <dict>
         <key>PayloadType</key>
         <string>com.apple.vpn.managed.applayer</string>
         <key>PayloadDescription</key>
         <string>Configures VPN settings, including authentication.</string>
         <key>PayloadDisplayName</key>
         <string>VPN (Per-App VPN App Mapping)</string>
         <key>PayloadIdentifier</key>
```

```
        <string>com.f5.access.macos.perapp.vpn.vpn</string>
        <key>PayloadOrganization</key>
        <string/>
        <key>PayloadUUID</key>
        <string>5A015006-D559-4C5C-B197-737CF4DCFA96</string>
        <key>PayloadVersion</key>
        <integer>1</integer>
        <key>Proxies</key>
        <dict/>
        <key>UserDefinedName</key>
        <string>App Mapping</string>
        <key>VPN</key>
           <dict>
              <key>AuthName</key>
              <string>test</string>
              <key>AuthPassword</key>
              <string>test</string>
              <key>AuthenticationMethod</key>
              <string>Password</string>
              <key>RemoteAddress</key>
              <string>https://portal.example.com</string>
              <key>OnDemandMatchAppEnabled</key>
              <true/>
              <key>ProviderType</key>
              <string>packet-tunnel</string>
              <key>ProviderBundleIdentifier</key>
              <string>com.f5.access.macos.PacketTunnel</string>
           </dict>
        <key>VPNSubType</key>
        <string>com.f5.access.macos</string>
        <key>VPNType</key>
        <string>VPN</string>
        <key>VendorConfig</key>
        <dict/>
        <key>VPNUUID</key>
        <string>6A015006-D559-4C5C-B197-737CF4DCFA96</string>
      </dict>
    </array>
    <key>PayloadDescription</key>
    <string>PerApp VPN Payload TCP Test</string>
    <key>PayloadDisplayName</key>
    <string>MDM - Per-App VPN TCP</string>
    <key>PayloadIdentifier</key>
    <string>com.f5.access.macos.perapp.vpn</string>
    <key>PayloadOrganization</key>
    <string/>
    <key>PayloadRemovalDisallowed</key>
    <false/>
    <key>PayloadType</key>
    <string>Configuration</string>
    <key>PayloadUUID</key>
    <string>06A850CC-BC81-43FB-AA16-42BE472D2421</string>
    <key>PayloadVersion</key>
    <integer>1</integer>
   </dict>
</plist>
```

Additional Access Policy Manager Configuration Information

F5 Access for macOS session variables

The following table contains a list of session variables and their attributes.

Session variable	Description
session.client.type	Indicates the client type, for example `Standalone`.
session.client.activex	Indicates whether ActiveX is supported. The result is always `0` for macOS.
session.client.platform	Indicates the platform type, such as `MacOS`.
session.client.plugin	Indicates whether the client is a plugin. This is always set to `0`.
session.client.app_id	The app ID for the client. For F5 Access for macOS this is `f5_access`.
session.client.app_version	The app version. For F5 Access for macOS 2.0.0 this is `2.0`.
session.client.model	Indicates the model name of the mobile device. For example, `MacBookPro`
session.client.platform_version	Indicates the platform and version of the mobile device. For example, Version 10.12.6 (Build 16G29)
session.client.jailbreak	Indicates the jailbreak status of the device. `0` indicates the device is not jailbroken, `1` indicates the device is jailbroken, and an empty response indicates that the status of the device is unknown.
session.client.cpu	Indicates the client CPU type. For example, `ARM`.
session.client.biometric_fingerprint	Indicates whether the device supports biometric fingerprint authentication. This is always set to `0` on macOS.
session.client.vpn_scope	Indicates the scope of the VPN tunnel. The result is `device` for a device-wide VPN connection, and `per-app` for a per-app VPN.
session.client.vpn_tunnel_type	Indicates the type of VPN tunnel. For F5 Access for macOS, this is `L3`.
session.client.vpn_start_type	Indicates how the VPN connection was initiated. • `manual` - Indicates that the connection was initiated by the user. • `on-demand` - Indicates that the connection was initiated by Safari Domains or Per-App VPN.
session.client.version	Indicates the client protocol version. For macOS the value is always `2.0`.
session.client.always_connected_mode	Indicates whether Always-On Mode is configured for the device. The result is always `0` for macOS.

Session variable	Description
session.client.hostname	This is the device host name (for example, `macos-system`).
session.client.js	Indicates whether the device used Web Logon mode. This is set to `0` for native logon, and `1` for web logon.
session.client.mdm_device_unique_id, session.client.unique_id	This value is provided by an MDM with the *`MdmDeviceUniqueId`* or *`UDID`* attribute. If both attributes are provided, *`MdmDeviceUniqueId`* takes preference. If neither is provided this session variable is not present. If this field is provided by the MDM, both session variables are present. An example value is `RC1KQLCJFOJEEM0XIOB3P52OMUQ3UN9Y3SDA5RWR`.
session.client.mdm_assigned_id	This value is provided by the MDM in the *`MdmAssignedId`* attribute. If this attribute is not provided, the session variable is not present.
session.client.mdm_instance_id	The value is provided by the MDM in the *`MdmInstanceId`* attribute. If this attribute is not provided, the session variable is not present.
session.client.mdm_device_wifi_mac_address	The value is provided by the MDM in the *`MdmDeviceWifiMacAddress`* or *`WiFiMAC`* attribute. If both attributes are provided, *`MdmDeviceWifiMacAddress`* takes preference. If neither attribute is provided, the session variable is not present.
session.client.mdm_device_serial_number	The value is provided by the MDM in the *`MdmDeviceSerialNumber`* or *`SerialNumber`* attribute.If both attributes are provided, *`MdmDeviceSerialNumber`* takes preference. If neither attribute is provided, the session variable is not present.

Index

V