F5 Access for iOS: Migration from 2.1.x to 3.x

1.0

Table of Contents

Changes in Virtual Servers with F5 Access

Virtual server changes for F5 Access

HTTP virtual server changes

If you currently use an HTTP virtual server, the connection to such a server is no longer supported due to Apple Transport Security (ATS) changes. Reconfigure the virtual server to use HTTPS.

HTTPS virtual servers and Apple Transport Security (F5 Access 3.x and 2.1.2 and later)

Because of Apple Transport Security changes, HTTPS requires the strongest TLS configuration (TLS 1.2 and PFS cipher suites). You may need to change the server certificate and the cipher settings in the clientssl profile to meet security requirements. All authentication ndpoints should comply with Apple's ATS requirements and use HTTPS TLS connections that comply with the following best practices:

- Use HTTPS with the strongest TLS configuration (TLSv1.2 with perfect forward secrecy cipher suites)
- Avoid using known-insecure cryptographic primitives (RC4 encryption and SHA-1 certificate signatures)
- Enforce key size requirements (2048 bits for RSA and 256 bits for EC)
- This includes the BIG-IP APM Client SSL profile and any external federated authentication providers (SAML IdP, Identity Provider or OAuth AS, Authorization Server).

The following cipher suites are supported:

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256

Virtual servers with Per-App VPN

If you use a virtual server for Per-App VPN connections, the `Application Tunnels (Java & Per-App VPN)` option is no longer required for Per-App VPN connections. However, this option does not need to be disabled. You can leave this setting enabled if you support both 2.1.x and 3.x clients on the same virtual server.

Access policy changes for F5 Access

About access policy changes for F5 Access

Client certificate authentication changes

F5 Access 3.x supports authentication with a client certificate in Web Logon mode on iOS 12 devices. This feature is supported on iOS 12 devices, but not on iOS 11 devices. However, in native logon mode, client certificate authentication is supported on both iOS 11 and iOS 12 devices.

To solve this, use native mode if possible in your deployment.

Restriction: *For F5 Access 3.x, native mode can be enforced in the Connectivity Profile on the BIG-IP. Please refer to the guide* ***BIG-IP APM and F5 Access for iOS*** *for details. This setting is available on BIG-IP 12.1.3, 13.1.0, and 14.0.0. This setting is note available on 11.5.1, 11.5.7, or 11.6.3.*

If you cannot use native mode, create different branches for iOS 12 and iOS 11 devices and use certificate authentication only on the iOS 12 branch. You can create a custom version check, as shown in the following example. Use the custom expression `expr {[mcget{session.client.platform.version}] >="12.0"}` to detect iOS 12 or later.

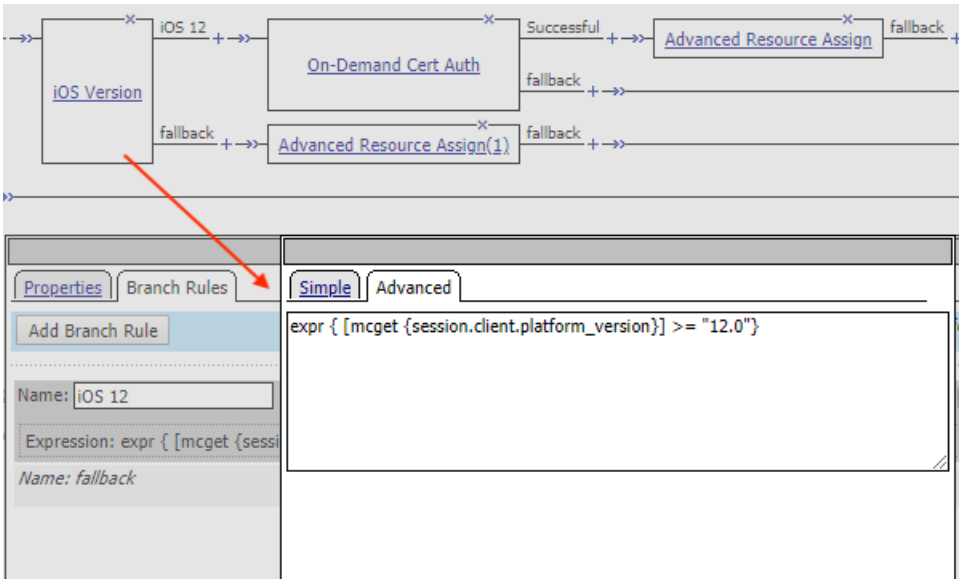


Figure 1: Version check for iOS 12

Changes with external logon pages

If you use an external logon page, that page must meet the Apple Transport Security (ATS) requirements, as detailed previously.

Client Proxy Autoconfig file changes

If you use a Client Proxy Autoconfig (PAC) script in your Network Acess, configuration, the file must be hosted on an HTTPS resource that meets Apple Transport Security (ATS) requirements, as detailed previously.

Device-wide On-Demand connections

If you use device-wide On-Demand connections, such connections now support runtime prompts and Web Logon connections with F5 Access 3.x. In the scenario where you have device-wide connections (but not per-app VPN connections), both manual connections and On-Demand connections can use the same Access Policy. Prompts that appear during authentication are supported, including password prompts, device authentication prompts, and Web Logon connections.

About Per-App VPN changes for F5 Access

Per-App VPN changes

Per-App VPN is a layer-3 tunnel in F5 Access 3.x. For the connection to work, a Network Access resource and a Webtop resource must be assigned to the Access Policy.

Per-app VPN connections do not fully support runtime prompts (password prompts, device authentication prompts) or Web Logon connections. We recommend that you configure the Access Policy so clients are not required to do interactive authentication in a Per-App VPN scenario.

You can use the session variable `session.client.vpn_scope` to identify device-wide and Per-App VPN connections.

Restriction: *This session variable can be used on BIG-IP versions 12.1.3, 13.1.0, and 14.0.0. This can not be used on 11.5.1, 11.5.7, or 11.6.3, as the session variable does not exist on those versions.*

- For the device-wide VPN branch, use `expr { [mcget {session.client.vpn_scope}] == "device" }`
- For the Per-App VPN branch, use `expr { [mcget {session.client.vpn_scope}] == "per-app" }`

See the following example.

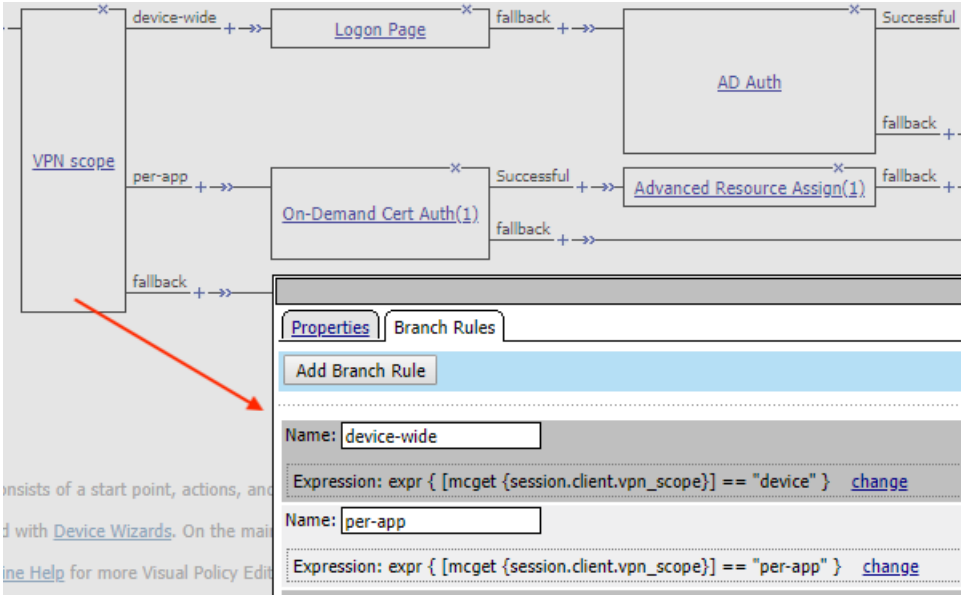


Figure 2: Per-App VPN access policy check

Adding a version check to the access policy

A version check allows you to distinguish between F5 Access for iOS 3.0.x and earlier versions. You can use this information to assign the required full network access resource to the 3.0.x branch, for example, in a Per-App VPN scenario.

Restriction: *This version check can be used on BIG-IP versions 12.1.3, 13.1.0, and 14.0.0. This can not be used on versions 11.5.1, 11.5.7, or 11.6.3, as the session variable does not exist on those versions.*

1. On the Main tab, click **Access** > **Profiles / Policies**.
   The Access Profiles (Per-Session Policies) screen opens.
2. In the Per-Session Policy column, click the **Edit** link for the access profile you want to configure.
   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 **Endpoint Security (Server-Side)** tab.
6. Select the **Client Type** item, and click **Add Item**.
7. Click **Save**.
8. On the Edge Client branch, click the **(+)** sign to add a new action item.
9. Click the **Endpoint Security (Server-Side)** tab.
10. Select the **Client OS** item, and click **Add Item**.
11. Click **Save**.
12. On the iOS branch, click the **(+)** sign to add a new action item.
13. Click the **General Purpose** tab.
14. Select the **Empty** item, and click **Add Item**.
15. On the Properties screen in the **Name** field, type `F5 Access Version`.
16. Click the Branch Rules tab.
17. Click **Add Branch Rule**.
18. In the Name field, type `Version 3`.
19. Click the **change** link in the Expression area.
    A popup screen opens.
20. Click the Advanced tab.

    Use this tab to enter Tcl expressions.

    A text input field displays.
21. In the text field, type `expr { [mcget {session.client.app_version}] >= "3.0" }`, and click Finished.
22. On the **fallback** branch following the F5 Access version item, change the Deny ending to Allow.

    iOS 2.x clients will take the fallback branch.
23. Click **Save**.
24. Add a Network Access resource to the `Version 3` branch. On the `Version 3` branch, click the **(+)** sign to add a new action item.

25. Click the **Assignment** tab.
26. Select the **Advanced Resource Assign** item, and click **Add Item**.
27. Under Resource Assignment, click **Add new entry**.
28. Under Expression, click **Add/Delete**.
29. Click the **Network Access** tab, and select a Network Access resource to assign.
30. Click the **Webtop** tab, and select a webtop to assign.
31. Click **Update**.
32. Click **Save**.
33. On the **fallback** branch following the Advanced Resource Assign item, click the Deny ending.
34. Change the Deny ending to Allow, and click **Save**.
35. Click **Apply Access Policy** to save your configuration.

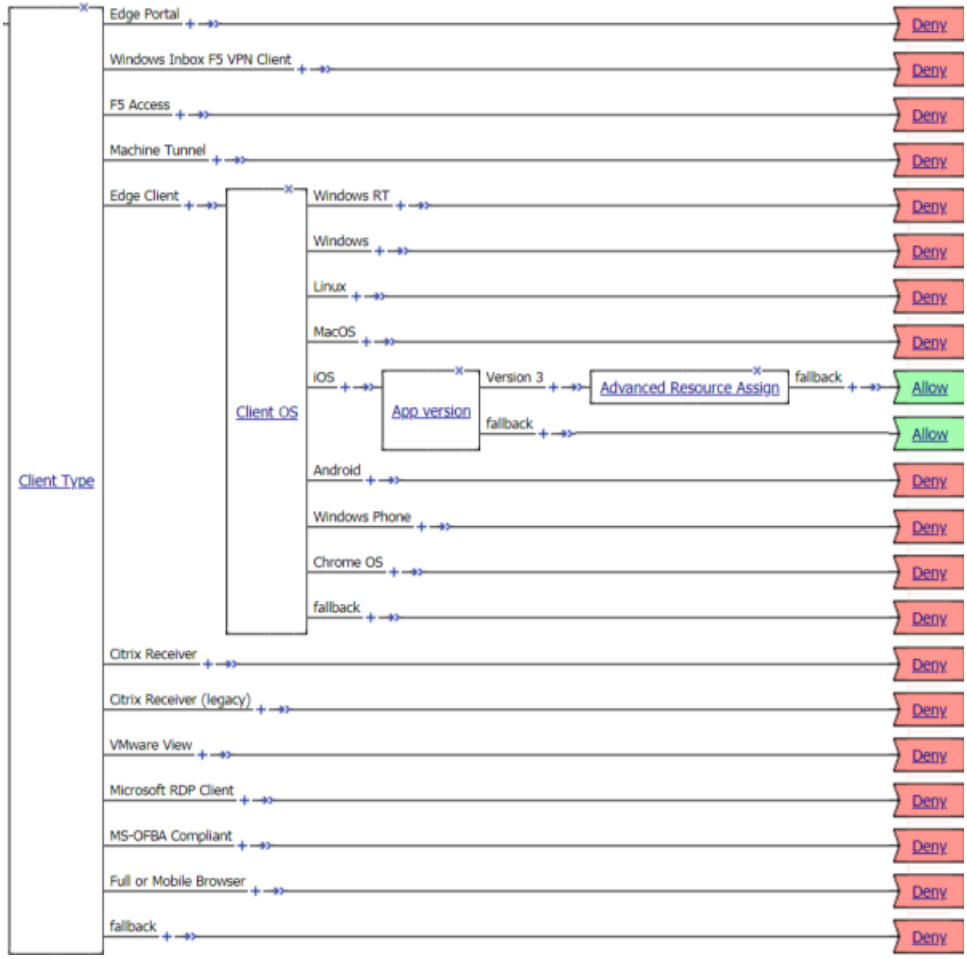
The access profile appears in the Access Profiles List.

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

Example of access policy for F5 Access 3.x and 2.1.x

You can configure an access policy branch to direct F5 Access 3.x and 2.1.x device users on iOS to different branches.

This example displays such an access policy.


Client Type
Edge Portal
Deny
Windows Inbox F5 VPN Client
Deny
F5 Access
Deny
Machine Tunnel
Deny
Edge Client
Client OS
Windows RT
Deny
Windows
Deny
Linux
Deny
MacOS
Deny
iOS
App version
Version 3
Advanced Resource Assign
fallback
Allow
fallback
Allow
Android
Deny
Windows Phone
Deny
Chrome OS
Deny
fallback
Deny
Citrix Receiver
Deny
Citrix Receiver (legacy)
Deny
VMware View
Deny
Microsoft RDP Client
Deny
MS-OFBA Compliant
Deny
Full or Mobile Browser
Deny
fallback
Deny

Client changes for F5 Access

About client changes for F5 Access

VPN Configurations do not migrate

VPN configurations created in F5 Access 2.1.x do not migrate to F5 Access 3.x. This applies to both manually created VPN configurations and configurations deployed with an MDM or with `.mobileconfig` files.

- For manually created VPN configurations, users must recreate the VPN configurations manually in F5 Access 3.x.
- For VPN configurations deployed with an MDM or `.mobileconfig` files, device-wide and Per-App VPN configurations deployed for F5 Access 2.1.x will not work on F5 Access 3.x. These configurations need to be re-deployed using updated VPN MDM profiles. See guidance on how to create VPN MDM profiles for F5 Access 3.x in the Managing Devices chapter, and in the Guide **BIG-IP APM and F5 Access for iOS**.

Changes with client certificates

All certificates that are installed in F5 Access 2.1.x are not used with F5 Access 3.x. This applies to certificates installed manually or with MDM or `.mobileconfig` files.

Note: *To access the user guide outside of the device, refer to F5 Access User Guide.*

- If a client certificate was manually installed by the user, the certificate must be imported again into F5 Access 3.x, using the new procedure, as described in the F5 Access User Guide on the device. Certificates in the system certificate storage are no longer used.
- If client certificates were installed with an MDM or using a `.mobileconfig` file, such certificates must be reinstalled with the new VPN MDM profile. See information on how to create these VPN MDM profiles for F5 Access 3.x in the Managing Devices chapter, and in the Guide **BIG-IP APM and F5 Access for iOS**.

Notifications

F5 Access 3.x prompts users to allow notifications. It is important that the user **Allow** these notifications if your deployment presents any prompts to user, including native prompts for username and password, Web Logon prompts, and device-authentication prompts. If notifications are not allowed, these scenarios cannot complete.

Device identity information

Because of changes with iOS, in F5 Access 3.x there is no method to obtain the UDID from the device. The session variable `session.client.mdm_device_unique_id` is submitted during authentication, if the value for this session variable is provided in an MDM profile.

Restriction: *The variable `session.client.mdm_device_unique_id` is submitted only on BIG-IP version 13.1.0 and later. This variable is not submitted on 11.5.1, 11.5.7, 11.6.3, or 12.1.3.*

For the purpose of backwards compatibility, the same value will be submitted as `session.client.unique_id` too, but again, only if this value is defined by the MDM profile.

Note: *This variable is submitted on all versions (11.5.1 through 14.1.0).*

If the device is not enrolled with an MDM, then no value for this variable is submitted. See information on how to create VPN MDM profiles for F5 Access 3.x in the Managing Devices chapter, and in the Guide **BIG-IP APM and F5 Access for iOS**.

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 custom device-wide VPN MDM profile

Your MDM may not currently support F5 Access for iOS 3.x. The VPN MDM profile for previous versions of F5 Access is not compatibile with F5 Access for iOS 3.x. If your MDM allows you to create custom configuration profiles, use these generic settings to configure the profile.

Important: *Consult with your MDM vendor to determine support. Refer to your MDM documentation before making changes.*

1. Add a VPN profile.
2. For the **Connection Type**, specify `Custom`.
3. For the **Identifier**, specify `com.f5.access.ios`.
4. Complete the rest of the configuration as required.

Creating a custom Per-App VPN MDM profile

Your MDM may not currently support F5 Access for iOS 3.x. The VPN MDM profile for previous versions of F5 Access is not compatibile with F5 Access for iOS 3.x. If your MDM allows you to create custom configuration profiles, use these generic settings to configure the profile.

Important: *Consult with your MDM vendor to determine support. Refer to your MDM documentation before making changes.*

1. Add a VPN profile.
2. For the **Connection Type**, specify `Custom`.
3. For the **Identifier**, specify `com.f5.access.ios`.
4. For the **Provider Type**, specify `Packet Tunnel`.
5. Complete the rest of the configuration as required.

Creating a configuration profile for the managed device

Before you assign a configuration profile to a device, that device must be enrolled with your MDM. Additionally, F5 Access must be installed on the 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 iOS

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 iOS UDID here</string>
```

```
        <key>MdmDeviceWifiMacAddress</key>
        <string>device wifi mac address here</string>
        <key>MdmDeviceSerialNumber</key>
        <string>device serial number here</string>
    </dict>
...
```

Web Logon setting

This setting configures Web Logon mode in an MDM profile.

Web Logon configuration

In the MDM configuration profile, you can use the command WebLogon to specify whether Web Logon is enabled. Use the syntax `<key>WebLogon</key><string>true|false</string>`.

If you configure Enforce Logon Mode in the Connectivity Profile on Access Policy Manager, that setting overrides the Web Logon setting configured in the MDM profile, or in a manual configuration. This setting is available on BIG-IP 12.1.3, 13.1.0, and 14.0.0. This setting is note available on 11.5.1, 11.5.7, or 11.6.3.

Note: *Web Logon is not supported with Per-App VPN.*

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 3.0 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.ios`
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</string>
         <key>PayloadDisplayName</key>
         <string>VPN</string>
         <key>PayloadIdentifier</key>

<string>com.apple.vpn.managed.E3C31113-0AC1-4085-BD77-6315F2ADA1EE</string>
         <!-- F5 COMMENT: PayloadType key: for System-Wide VPN
         the value is "com.apple.vpn.managed" -->
         <key>PayloadType</key>
         <string>com.apple.vpn.managed</string>
         <key>PayloadUUID</key>
         <string>E3C31113-0AC1-4085-BD77-6315F2ADA1EE</string>
         <key>PayloadVersion</key>
         <integer>1</integer>
         <key>Proxies</key>
         <dict>
            <key>HTTPEnable</key>
            <integer>0</integer>
            <key>HTTPSEnable</key>
            <integer>0</integer>
         </dict>
         <key>UserDefinedName</key>
         <string>VPN Config</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://demo-na-bigip.com</string>
         </dict>
         <!-- F5 COMMENT: VPNSubType key: For F5 Access the value
         should be "com.f5.access.ios" -->
         <key>VPNSubType</key>
         <string>com.f5.access.ios</string>
         <!-- F5 COMMENT: VPNType key: Specifies VPN type,
         for F5 Access VPN should be "VPN" -->
         <key>VPNType</key>
         <string>VPN</string>
         <key>VendorConfig</key>
         <dict/>
      </dict>
   </array>
   <key>PayloadDisplayName</key>
   <string>SystemwideVPNDemo</string>
```

```
   <key>PayloadIdentifier</key>
   <string>XYZ-ML-00003638.DBCD844F-1B48-55AF-A262-82B10131000D</string>
   <key>PayloadRemovalDisallowed</key>
   <false/>
   <key>PayloadType</key>
   <string>Configuration</string>
   <key>PayloadUUID</key>
   <string>842BF859-9305-4E86-A73F-8C44E1E36D72</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.ios`
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. 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

Key	Type	Description
		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</string>
         <key>PayloadDisplayName</key>
         <string>VPN</string>
         <key>PayloadIdentifier</key>

<string>com.apple.vpn.managed.CF2C73E8-B7AD-442F-BF91-2682777023CC</string>
         <!-- F5 COMMENT: PayloadType key: for Per-App VPN the value
         is "com.apple.vpn.managed.applayer" -->
         <key>PayloadType</key>
         <string>com.apple.vpn.managed.applayer</string>
         <key>PayloadUUID</key>
         <string>CF2C73E8-B7AD-442F-BF91-2682777023CC</string>
         <key>PayloadVersion</key>
         <integer>1</integer>
         <key>Proxies</key>
         <dict>
            <key>HTTPEnable</key>
            <integer>0</integer>
            <key>HTTPSEnable</key>
            <integer>0</integer>
         </dict>
         <key>UserDefinedName</key>
         <string>Per-App VPN Demo</string>
         <key>VPN</key>
         <dict>
            <key>AuthName</key>
            <string>username</string>
```

```
            <key>AuthPassword</key>
            <string>password</string>
            <key>AuthenticationMethod</key>
            <string>Password</string>
            <!-- F5 COMMENT: ProviderType key: F5 Access 3.x supports
            only "packet-tunnel" value for this key -->
            <key>ProviderType</key>
            <string>packet-tunnel</string>
            <key>OnDemandMatchAppEnabled</key>
            <true/>
            <key>RemoteAddress</key>
            <string>https://demo.siterequest.com</string>
         </dict>
          <!-- F5 COMMENT: VPNUUID key: A globally-unique identitifer
          for the VPN configuration. This identifier is used to configure
          apps so that they use the Per-App VPN service for
          all of their network communication -->
         <key>VPNUUID</key>
         <string>17027186-61c3-470d-afaa-5a9e4d519da1</string>
         <!-- F5 COMMENT: VPNSubType key: For F5 Access the value
         is "com.f5.access.ios" -->
         <key>VPNSubType</key>
         <string>com.f5.access.ios</string>
         <!-- F5 COMMENT: VPNType key: Specifies VPN type,
         for F5 Access VPN is "VPN" -->
         <key>VPNType</key>
         <string>VPN</string>
         <key>VendorConfig</key>
         <dict/>
         <key>SafariDomains</key>
         <array>
            <string>test.siterequest.com</string>
         </array>
      </dict>
   </array>
   <key>PayloadDisplayName</key>
   <string>PerAppVPNDemo</string>
   <key>PayloadIdentifier</key>
   <string>XYZ-ML-00003638.C4B7F07B-9C1C-F3F2-BB80-A30390AD085F</string>
   <key>PayloadRemovalDisallowed</key>
   <false/>
   <key>PayloadType</key>
   <string>Configuration</string>
   <key>PayloadUUID</key>
   <string>BD56E80E-BFCE-4FD6-AEDB-543014C6ADE8</string>
   <key>PayloadVersion</key>
   <integer>1</integer>
</dict>
</plist>
```

Index