BIG-IP® APM and F5 Access for iOS

Version 3.0.1

Table of Contents

Overview: F5 Access for iOS

Introducing F5 Access 3.x

F5 Access for iOS 3.x is a new client, built on the latest Apple VPN architecture. Apple's new Network Extension architecture allows for some features that were not previously included in our iOS client, including the ability to use UDP apps with Per-App VPN. Apple has deprecated their previous VPN technology, which will not be supported in the future, so our previous clients based on older technology will eventually be deprecated as well.

This is not a one-to-one upgrade from the previous version (F5 Access 2.x). A number of incompatibilities, possible incompatibilities, and configuration changes are outlined in this document that may affect your migration to F5 Access for iOS 3.x. MDM support for this new client is still in development. Please check with your MDM vendor for more information.

There are access policy changes required to support this client. If you are planning to migrate users to the new client, please review all of the differences between the clients outlined in this document before you migrate your users. We expect to add features and to support to this client in the future, and eventually we expect the same level of support from MDM vendors with our existing client.

Note: *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.*

Differences between F5 Access 3.x and F5 Access Legacy 2.1.x

There are a number of differences between F5 Access 3.x and F5 Access Legacy 2.1.x.

Configuration deployment changes

When deploying configurations, there are several differences between F5 Access 3.x and F5 Access Legacy 2.1.x.

Table 1: Deployment differences

VPN type	Manually configured	MDM configured
Device-wide VPN	User has to accept a permission dialog to add the first VPN configuration	The key `VPNSubType` has changed. • In F5 Access Legacy 2.1.x: `com.f5.F5-Edge-Client.vpnplugin` • In F5 Access 3.x: `com.f5.access.ios`

Per-App VPN	No manual configuration	• The key `VPNSubType` has changed:
		  • In F5 Access Legacy 2.1.x: `com.f5.F5-Edge-Client.vpnplugin`
		  • In F5 Access 3.x: `com.f5.access.ios`

VPN type	Manually configured	MDM configured
		• The key *ProviderType* must be set to *packet-tunnel* in F5 Access 3.x. • The key *PerAppVpn* is no longer required in the VendorConfig dictionary in F5 Access 3.x.

Device UDID change

Device UDID is no longer provided, due to iOS changes. With an MDM, the device can be assigned an ID. This is assigned with the *MdmDeviceUniqueId* or *UDID* attribute. This assigned value populates the session variables `session.client.mdm_device_unique_id` and `session.client.unique_id`. If neither is provided this session variable is not present. If either field is provided by the MDM, both session variables are present. An example value is `RC1KQLCJFOJEEM0XIOB3P52OMUQ3UN9Y3SDA5RWR`.

VPN establishment changes

When establishing VPNs, there are several differences between F5 Access 3.x and F5 Access Legacy 2.1.x.

Table 2: VPN establishment changes

VPN type	Manual	On-demand
Device-wide VPN	• In F5 Access 3.x, notifications must be enabled for any user prompts or Web Logon interactions. • In F5 Access 3.x, the user is able to save the password when connecting in native logon mode if the **Save Password Method** option in the Access Policy Manager Connectivity Profile is set to `disk`.	In F5 Access 3.x, notifications must be enabled for any user prompts or Web Logon interactions. With notifications enabled, these prompts and features are supported. • Web Logon mode • Authentication prompts in native mode • Device authentication
Per-App VPN	No manual configuration	A Per-App VPN connection cannot be established if user interaction is required. For F5 Access 3.x, configure the access policy so user interaction is not required to establish the VPN connection.

Access Policy Manager configuration changes

When configuring Access Policy Manager, there are several differences between F5 Access 3.x and F5 Access Legacy 2.1.x.

Table 3: Enforcing logon mode

APM configuration item	Change
Enforce Logon Mode	In the Connectivity Profile, the administrator can now enforce a specific logon mode, using the setting **Enforce Logon Mode**. The logon mode can be enforced as *native* or *web*.

APM configuration item	Change
Web Logon mode in F5 Acesss for iOS app	If **Enforce Logon Mode** is enabled in the Connectivity Profile, the user cannot change the Web Logon option.

Table 4: APM Per-App VPN changes

Per-App VPN configuration item	Change
Virtual Server	In the Virtual Server configuration, the option **Application Tunnels (Java & Per-App VPN)** is no longer required to be enabled
Access policy	With F5 Access 3.x, Per-App VPN now uses an L3 tunnel. As such, the following items must be added to the applicable access policy branch: • Network Access resource • Webtop
iOS device	The iOS device enforces the applications that are allowed to access the VPN, according to the Per-App VPN configuration.

Apple App Transport Security (ATS) changes

Apple Transport Security (ATS), implemented in F5 Access 3.x, requires the following security changes for communications between F5 Access 3.x and the corresponding BIG-IP.

- Plain text HTTP connections are no longer allowed.
- HTTPS requires the strongest TLS configuration (TLS 1.2 and PFS cipher suites).
- Self-signed certificates are not supported unless the CA certificate is first Trusted on the device.

Note: *These Apple Transport Security changes are also required for Web Logon connections with F5 Access legacy 2.1.x.*

Client Certificate authentication

Client Certificate Authentication is not supported in Web Logon mode on iOS 11. On iOS 12, Web Logon mode does support Client Certificate Authentication.

F5 Access and mobile devices

F5 Access for mobile devices 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 mobile devices.

For information about how to use F5 Access on your device, refer to the *F5 Access for iOS User Guide*.

F5 Access features include:

- N-factor authentication (at least two input fields, password and passcode) support
- User name and password, client certificate, and RSA SecurID support
- Multiple input field support

- Credential caching support
- Support for TouchID authentication, PIN, or a device password to make a connection, when using cached credentials
- Support for DNS address space for split-tunneling configurations
- Support for checking information from client devices
- Support for automatically launching applications on client devices
- Support for roaming between cellular and WiFi networks
- Landing URI support
- Logging support to report issues
- Support for private-side internal proxy servers. Public-side proxy servers are not currently supported.
- Per-app VPN support for TCP and UDP applications
- Application notifications
- Diagnostics
- Traffic Graphs
- Support for SAML 2.0 features in BIG-IP® Access Policy Manager®
- iOS widget support

About app notifications

F5 Access for iOS 3.x requires that notifications be enabled for most user configurations. This requires that the app be started by the user and accept notifications.

Important: *The user is prompted to enable notifications only the first time the app is started. After the first app start, if the notifications dialog is dismissed, the user must manually enable notifications. If the user dismisses the notification dialog, the user can enable notifications manually. To enable notifications, in the* ***Settings*** *app, go to* ***F5 Access*** *>* ***Notifications****, and enable the* ***Allow Notifications*** *setting.*

Note: *Notifications are not required to be enabled, only in a Per-App VPN scenario where no user intervention is required.*

About SAML support

F5 Access for iOS devices provides the following SAML support:

- Service provider-initiated access only, for example, APM acting as the service provider (SP)
- Web Logon mode only
- Single Log-Out (SLO): supported only when the logout action is initiated from the client

When you use F5 Access as a client performing SP-initiated access, F5 Access first connects to BIG-IP® Access Policy Manager® (APM®). Because there is no assertion, APM redirects the client to the IdP. The IdP then authenticates the user and redirects F5 Access back to the SP with assertion. APM then accepts the assertion and establishes a VPN connection. You can then access back-end resources through >F5 Access.

You can configure a BIG-IP system by configuring APM as an SP. The access policy that is associated with the configuration assigns a SAML AAA resource followed by a Network Access Resource. For more information about SAML configurations, refer to the *BIG-IP® Access Policy Manager®: Authentication and Single Sign-On* guide.

About supported authentication types

F5 Access for iOS 3.x supports these authentication and connection type combinations.

Tip: *You can create a .mobileconfig file with Apple Configurator 2. Read Apple Configurator 2 documentation for more information.*

Authentication type	Connection type
Username and password	Runtime prompts (login dialogs, device authentication, 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 For a Per-App VPN connection, runtime prompts are not supported, so the username and password must be specified in device configuration specified by the MDM, or in the .mobileconfig file. Per-App VPN does not support Web Logon mode.
Client certificate	• User-initiated connections, in native mode only • Device-wide VPN On-Demand, in native mode only • Per-App VPN connections
Client certificate + username and password	Runtime prompts (login dialogs, device authentication, and other user input prompts) are allowed for: • User-initiated connections, in native mode only. • Device-wide VPN On-Demand connections, in native mode only. For a Per-App VPN connection, runtime prompts are not supported, so the username and password must be specified in the configuration. Per-App VPN does not support Web Logon mode.

About establishing VPN connections

The F5 Access application (app) for mobile 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.

For Per-App VPN, the following on demand considerations apply. These do not apply to On-Demand device-wide VPN connections.

- When a Per-App VPN connection is initiated On-Demand, user intervention is not allowed. For example, if a password is needed for authentication, but is not supplied in the configuration, the connection fails. Note that RSA authentication is not supported.
- On-Demand Per-App VPN does not work with Web Logon.

About pre-logon checks supported for iOS devices

Access Policy Manager® can check unique identifying information from an iOS client device. The supported session variables, which become populated with the iOS client device information, are gathered automatically, and can easily be combined with an LDAP or AD query to implement white-listing in a custom action to improve access context. This information allows Access Policy Manager to perform pre-logon sequence checks and actions based on information about the connecting device. Using such information, Access Policy Manager can perform the following tasks:

- Deny access if the iOS version is less than the required level.
- Deny access if the app version is less than required.

This example displays an access policy with a custom action to check the app version.

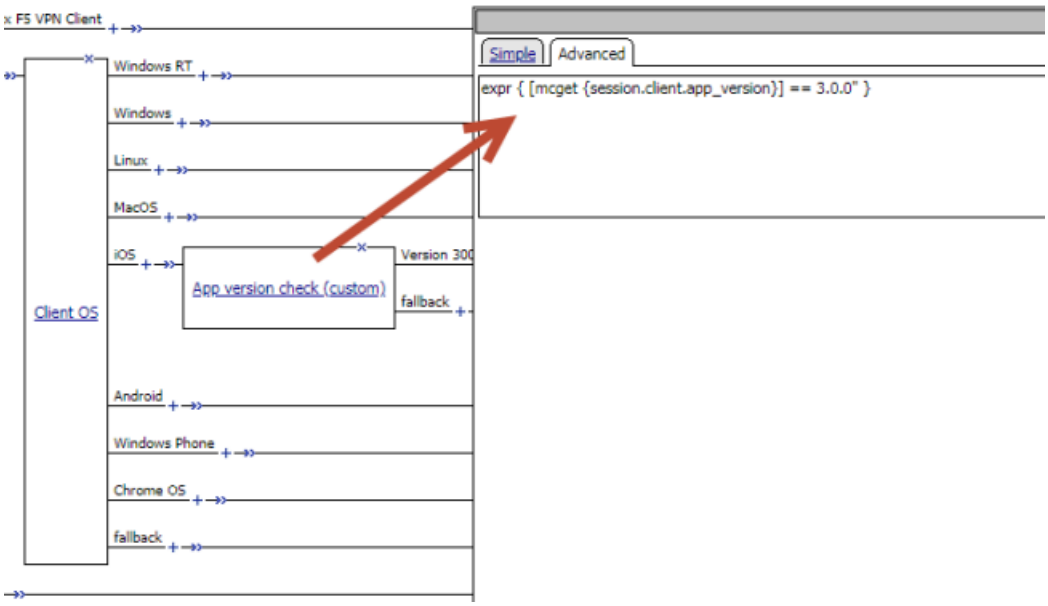


Figure 1: Example of a custom action for checking the F5 Access app version

About automatically launching applications from mobile devices

You can configure F5 Access to launch an app with a registered URL scheme after a VPN connection is established.

Auto-launching applications from F5 Access

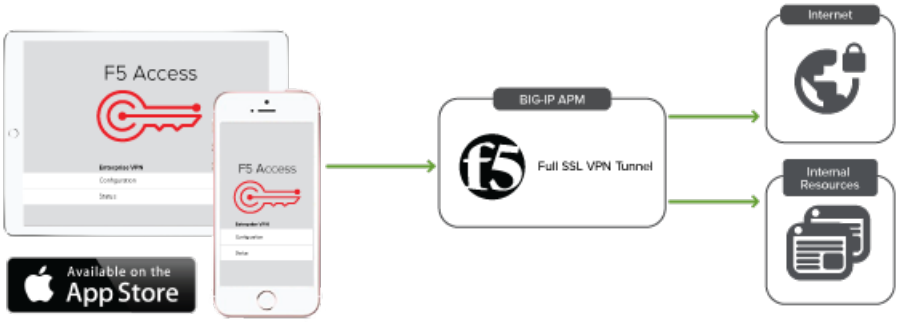
You can configure applications to automatically start on F5 Access once a connection is initiated.

1. On the Main tab, click **Access** > **Connectivity / VPN** > **Network Access (VPN)** > **Network Access Lists**.
2. Click the name of your network access resource on the list.
3. Click the **Launch Applications** tab.
4. Click **Add**.
5. In the **Application Path** field, type in your application path in the form of a URL scheme, for example, `skype://14082734800?call`.

6. Type any required parameters in the **Parameters** field.
7. From the **Operating System** list, select iOS.
8. Click **Finished.**
   On the device, a warning is issued before the local application executes.

About network integration on iOS devices

Access Policy Manager® provides web application-level security to prevent malware attacks. As an administrator, you can enforce all web access through a secured gateway, as well as bypass secure gateways for internal resources. This is especially helpful, for example, when you have clients using corporate tablets, smartphones, or other mobile devices to browse the web.



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 iOS and 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**.

Prerequisites for configuring F5 Access

Before configuring F5 Access for iOS 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 iOS devices

To configure F5 Access for iOS 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. In the Basic Properties area of the wizard, clear the **Enable Antivirus Check in Access Policy** check box for Client Side Checks to ensure that your users can connect with F5 Access.
4. Click **Finished**.

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

Customizing client proxy settings for iOS

Configure Network Access to provide further fucntionality for F5 Access connections.

1. On the Main tab, click **Access** > **Connectivity / VPN** > **Network Access (VPN)**.
   The Network Access List screen opens.
2. Select a Network Access resource to edit.
3. Select **Client Proxy Settings**.
4. If you want to use an optional client proxy autoconfig (PAC) script, in the **Client Proxy Autoconfig Script** field type the URL for a proxy auto config script.
5. If you want to use an optional client proxy address, in the **Client Proxy Address** field, type the IP address for the client proxy server that network access clients use to connect to the Internet.
6. If you want to use an optional client proxy port, in the **Client Proxy Port Type** type the port number on the proxy server that you want network access clients to use to connect to the Internet.
7. If you want to bypass some addresses with the client proxy, in the **Client Proxy Exclusion List** field specify the Web addresses that do not need to be accessed through the proxy server. You can use wild cards to match domain and host names or addresses. For example, **www.*.com**, **128.***, /, **8.**, **mygroup.***, and ***.***.

Customizing an access policy to support F5 Access on Access Policy Manager

Create an access policy that supports F5 Access for iOS.

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 plus **(+)** sign that appears before the `Logon Page` action.
4. Under **Server Side Checks**, select **Client Type**, and click **Add Item**.
5. Click **Save**.
   The Client Type action is added to the access policy, and several new branches appear.
6. On the Edge Client branch of the Client Type action, click the plus **(+)** sign.
7. Under **Server Side Checks**, select **Client OS**, and click **Add Item**.
8. Configure the **iOS** Branch Rule with the configuration objects and resources you want to assign to iOS F5 Access.
9. Click **Finished**, and then click **Save**.
10. Add the network access resource to the branch.
11. Click **Save**.
    This access policy now supports F5 Access for iOS.

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 iOS 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 iOS client will fail to the fallback branch of the first client-side check. Assign authentication and a network access resource to this branch.
- Add a **Client OS** Access Policy item, and assign authentication and resources to the **iOS** branch.

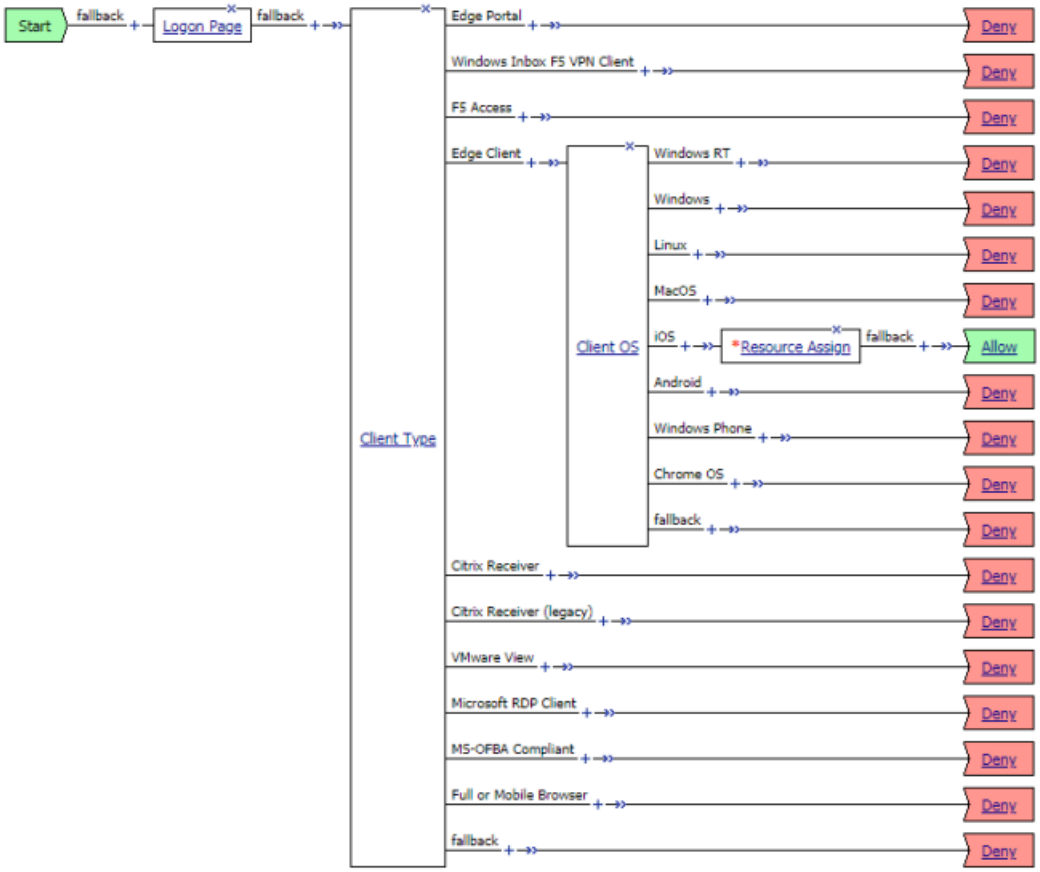
F5 Access for iOS is detected with the following access policy items:

Access policy item	Value
Client Type	Edge Client
Client OS	iOS

Example of basic access policy that supports F5 Access

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

This example displays a simple access policy.


Start
fallback
Logon Page
fallback
Client Type
Edge Portal
Deny
Windows Inbox F5 VPN Client
Deny
F5 Access
Deny
Edge Client
Client OS
Windows RT
Deny
Windows
Deny
Linux
Deny
MacOS
Deny
iOS
*Resource Assign
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

Configuring Per-App VPN with APM and F5 Access

What is per-app VPN?

Apple's Network Extension framework supports layer-3 tunneling for both device-wide and Per-App VPN tunnels. This means that TCP and UDP protocols are supported for apps configured for Per-App VPN on F5 Access for iOS 3.x. Apps that are managed by a Mobile Device Manager (MDM) can be configured to automatically connect to a VPN when they are started. In addition, Mobile Safari can be managed for per-app VPN with a configuration profile and without an MDM. Per-app VPN gives IT granular control over corporate network access, and ensures that data transmitted by managed apps travels only through a VPN. Meanwhile, other data, like an employee's personal web browsing activity, does not use the VPN. Per-app VPN also works with Safari on a per-URL basis.

A per-app VPN configuration requires three configuration components.

- A device under MDM management, or a configuration profile file installed manually. For more information, see *Configuration Profile Reference*.
- A managed app installed on the device, or Mobile Safari.
- F5 Access for iOS installed on the managed device.

Important: *The managed app and the MDM profile must be deployed with an MDM solution, except in the case of Mobile Safari. The F5 Access configurations may or may not be deployed with an MDM solution. Any app other than Mobile Safari must be installed by the MDM solution, and associated with a VPN configuration.*

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).

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 will use that VPN only.*

In this example, App 1 or App 2 can be active at the same time, because they use different VPN configurations.

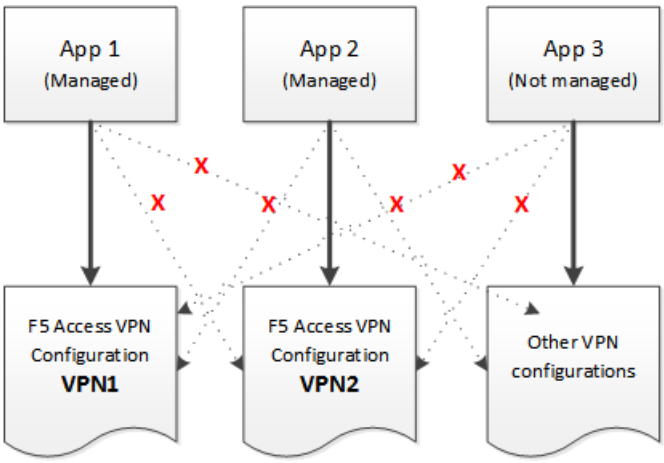


Figure 2: Apps associated with different VPN configurations

Note: *On iOS, you can only activate only one device-wide (user-initiated) VPN configuration at a time. However, multiple per-app VPNs can be active and connected simultaneously, on their own or in addition to the device VPN.*

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** > **Profiles / Policies**.
   The Access Profiles (Per-Session Policies) screen opens.
2. Click **Create**.
   The New Profile screen opens.
3. In the **Name** field, type a name for the access profile.
4. From the **Profile Type** list, select **SSL-VPN**.
5. 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.
6. Click **Finished**.

The access profile appears in the Access Profiles List.

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.

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 `iOS 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. Click **Save**.
23. Add a Network Access resource to the Version 3 branch. On the Version 3 branch, click the **(+)** sign to add a new action item.
24. Click the **Assignment** tab.
25. Select the **Advanced Resource Assign** item, and click **Add Item**.
26. Under Resource Assignment, click **Add new entry**.
27. Under Expression, click **Add/Delete**.
28. Click the **Network Access** tab, and select a Network Access resource to assign.
29. Click the **Webtop** tab, and select a webtop to assign.
30. Click **Update**.
31. Click **Save**.
32. On the **fallback** branch following the Advanced Resource Assign item, click the Deny ending.
33. Change the Deny ending to Allow, and click **Save**.
34. If you support F5 Access version 2.x clients, on the fallback branch, click the Deny ending.
35. Change the Deny ending to Allow, and click **Save**.
36. 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.

Adding a client certificate check to the access policy

A client certificate check allows you to authenticate the device to the access policy, without requiring any user interaction that would cause the creation of the per-app VPN tunnel to fail.

1. On the Main tab, click **Access** > **Profiles / Policies**.
2. In the Access 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, and click **Add Item**.
7. The properties screen opens. Click **Save**.
8. On the **Successful** branch following the Client Cert Inspection item, click the Deny ending.
9. Change the Deny ending to Allow, and click **Save**.
10. 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.

About setting up Access Policy Manager for per-app VPN

You configure specific settings in the Access Policy Manager® to provide per-app VPN tunnels. Per-app VPN tunnels are full network access tunnels, and require Network Access resources in the Access Policy. Configure these items on the Access Policy Manager.

- The virtual server must be configured with an access profile.
- The virtual server should be configured with a basic configuration for the network access resource.
- You must specify the Client SSL profile on the virtual server. You must also include the same CA bundle on the server that is used to generate the certificate for the client devices.

Note: *Access policies for F5 Access Legacy 2.1.x have different requirements. If you are planning to have both clients connect to the same virtual server, refer to your F5 Acccess 2.1.0 documentation for more information.*

Configuring a virtual server for per-app VPN

You must have Access Policy Manager® licensed and provisioned.

A virtual server profile enables support for the network access used by per-app VPN tunnels.

1. On the Main tab, click **Local Traffic** > **Virtual Servers**.

The Virtual Server List screen opens.

2. Click the name of the virtual server you want to modify.
3. For the **SSL Profile (Client)** setting, from the **Available** list, select the name of the Client SSL profile you previously created, and using the Move button, move the name to the **Selected** list.
4. In the Access Policy area, from the **Access Profile** list, select the access profile.
5. From the **Connectivity Profile** list, select the connectivity profile.
6. Click **Update** to save the changes.

The virtual server is configured for per-app VPN.

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 5: 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.

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 6: 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 7: 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>
```

Additional Access Policy Manager Configuration Information

F5 Access for iOS 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.platform	Indicates the platform type, such as `iOS`.
session.client.app_id	The app ID for the client. For F5 Access for iOS this is `com.f5.Edge-Client`.
session.client.app_version	The app version for the client. For F5 Access 3.0.1 this is `3.0.1`.
session.user.agent	Indicates the browser, device type, and operating system version of the client, as well as the version of F5 Access.
session.client.model	Indicates the model name of the mobile device. For example, `iPhone`
session.client.platform_version	Indicates the platform and version of the mobile device. For example, 11.1
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.biometric_fingerprint	Indicates whether the device supports biometric fingerprint authentication. `1` indicates that a fingerprint is configured, `0` indicates that a fingerprint is not configured, or the device does not support fingerprint authentication.
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 iOS, 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 connection is either a device-wide VPN triggered On-Demand or a Per-app VPN connection.
session.client.version	Indicates the client protocol version. For iOS, the value is always `2.0`.

Session variable	Description
session.client.device_passcode_set	Indicates whether the user has a device unlock passcode, PIN, or biometric authentication configured. The results is `1` if a device lock is configured, and `0` if it is not.
session.client.browscap_info	Specifies the browser information presented. For example, uimode=7&ctype=Standalone &cversion=2.0&cjs=0&cactivex=0 &cplugin=0&cplatform=iOS&cpu=ARM
session.client.hostname	This is the device host name (for example, `SandysiPhone`).
session.client.js	Indicates whether the device used Web Logon mode to log on. The result is `1` if Web Logon Mode was used, and `0` if it was not.
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.

Access Policy Manager configuration tips

The following table provides tips for setting up F5 Access for devices.

Feature	Information
Client endpoint checks	Client end-point checks are not currently supported.
Require Device Authentication	For devices with iOS 9 or later, F5 Access can require device authentication with one of the device locking methods, including biometric authentication (Touch ID), a PIN, or a passphrase. To enable device authentication for F5 Access, in the **Connectivity Profile** under **iOS Edge Client**, enable the options **Allow Password Caching** and **Require Device Authentication**.

Feature	Information

Password caching policy	• In the Connectivity profile, you can configure password caching by enabling the setting `Allow Password Caching`. When this setting is enabled, after a successful logon the submitted credentials are cached.
	• Specify a `Save Password Method`.
	  • If you select **disk**, an encrypted password is cached on the device 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.
	• Credentials are not cleared if the user disconnects or restarts the device.
	• If credentials are cached and the **Save Password Method** is **memory**, then credentials are cached until one of the following events occurs:
	  • The specified credential cache duration expires.
	  • The server address of the configuration within the application changes.
	  • The username of the configuration within the application changes.
	  • The F5Access user switches between configurations.
	• To require the user to authenticate on the device before unlocking the cached credentials, select **Require Device Authentication.**

Enforce Logon Mode	You can enforce the logon mode for the iOS client. In the Connectivity Profile, select **iOS Edge Client**, and click **Enforce Logon Mode**. Select **Native** or **Web** and click **OK**. The logon mode will be enforced for all clients that use the connectivity profile.
Client certificates	Client certificate authentication is supported, either with a certificate alone or with a certificate secured with a user name and password.
On-Demand Cert Auth	If used, the `On-Demand Cert Auth` action must be placed after other authentication actions in the access policy.

About starting the client from a URL scheme

You can start F5 Access connections for users from a URL. You can then provide these URLs to users, so they can start the VPN connection without having to manually start the application. If there is already an active connection, a prompt appears to warn the user that the existing connection must be stopped before the new connection can start. The connection uses a client certificate if it is specified in the existing configuration.

URL connections use the following parameters. This is an example, you must provide your own parameters and values.

```
f5access://{start|stop}?[parameter1=value1&parameter2=value2...]
```

Note: *Special characters in parameters must be URL-encoded.*

The syntax to start a connection from a URL follows.

start
Starts a connection. The `start` command requires either the `name` or `server` parameter to be present in the URL. If the `name` parameter is specified, then F5 Access looks for the name in the list of existing configuration entries. If the `server` parameter is specified, then the `name` parameter is set to the same value as the `server` parameter. A new configuration is created if a configuration with that name does not exist. If the specified configuration already exists, the other parameters specified in the URL are merged with the existing configuration. The result of this merged configuration is used only for the current, active connection, and does not persist. If a `name` is specified with other parameters, such as `server`, `username`, or `password`, those parameters override what is specified in the configuration.

username
A parameter used to specify the user name with which to start the connection. When the `username` is specified without a `password`, then an authentication prompt is displayed.

password
A parameter used to specify the password with which to start the connection. When the `password` parameter is specified, it is used as a one-time password and not saved in the configuration.

postlaunch_url
A parameter used to specify the URL that starts after the connection starts.

logon_mode
An optional parameter that specifies whether the logon mode is the standard logon (`native`) or web logon (`web`). The default logon mode is `native`.

Examples of starting a client from a URL

The following examples illustrate how to start F5 Access connections for users from a URL.

Connecting to an existing configuration called `MYVPN`:

```
f5access://start?name=MYVPN
```

Connecting to an existing configuration called `MYVPN` and including the server URL `myvpn.siterequest.com`:

```
f5access://start?name=MYVPN&server=myvpn.siterequest.com
```

Connecting to a specific server called `myvpn.siterequest.com`:

```
f5access://start?server=myvpn.siterequest.com
```

Connecting to a specific server called `myvpn.siterequest.com` with web logon enabled:

```
f5access://start?server=myvpn.siterequest.com&logon_mode=web
```

Connecting to an existing configuration called `MYVPN` and including the username `smith` and the password `passw0rd`:

```
f5access://start?name=MYVPN&username=smith&password=passw0rd
```

Starting a connection to a configuration called `MYVPN` and specifying the post-launch URL `jump://?host=10.10.1.10&username=smith`:

```
f5access://start?name=MYVPN&postlaunch_url=jump%3A%2F%2F%3Fhost%3D10.10.1.10
%26username%3Dsmith
```

Stopping a connection:

```
f5access://stop
```

About defining a server from a URL

You can add BIG-IP® server definitions to F5 Access from a URL. You can provide these URLs to users, so they can create and/or start VPN connections without having to manually start the application.

Use the following URL and parameters to create a server:

```
f5access://create?server=server_address[&parameter1=value1&parameter2=value2...]
```

Note: *Special characters in parameters must be URL-encoded.*

The syntax to define a server from a URL follows.

`server`
The server address is either a DNS name or an IP address.

`name`
An optional description of the server.

`username`
An optional parameter used to specify the user name with which to start the connection. When the `username` is specified without a `password`, then an authentication prompt is displayed. If no `username` is specified during server creation, the user is prompted for it at session initiation, if required.

`password`
An optional parameter used to specify the password with which to start the server connection. When the `password` parameter is specified, it is used as a one-time password and not saved in the configuration.

`logon_mode`

Specifies whether the logon mode is the standard logon (`native`) or web logon (`web`). The default logon mode is `native`.

`domain_never`
An optional, comma-separated list of match pattern(s) for the Never Connect domain list, for iOS devices only.

`domain_ifneeded`
An optional, comma-separated list of match pattern(s) for the Connect If Needed domain list, for iOS devices only.

Examples of defining a server from a URL

The following examples illustrate how to define servers for F5 Access connections from a URL.

Create a server at `edgeportal.siterequest.com`:

```
f5access://create?server=edgeportal.siterequest.com
```

Create a server named `EdgePortal` with the server URL `edgeportal.siterequest.com`:

```
f5access://create?name=EdgePortal&server=edgeportal.siterequest.com
```

Index

S

T

U

V

W