

**BIG-IP[®] Access Policy Manager[®] and
BIG-IP[®] Edge Client[™] for iOS v2.0.4
Technical Note**

Contents

Legal Notices.....	5
Acknowledgments.....	7
Chapter 1: Overview: BIG-IP Edge Client for Mobile Devices.....	11
What does BIG-IP Edge Client do for mobile devices?.....	12
About SAML support.....	12
About supported authentication types.....	12
About establishing VPN connections.....	13
About pre-logon checks supported for iOS devices.....	13
About automatically launching applications from mobile devices.....	14
Auto-launching applications from the BIG-IP Edge Client.....	14
About network integration on iOS devices.....	14
Setting up a network access.....	14
Chapter 2: Configuring Access Policy Manager for BIG-IP Edge Client.....	17
Access Policy Manager configuration for BIG-IP Edge Client for mobile devices.....	18
Running the Network Access Setup Wizard.....	18
Customizing an access policy to support BIG-IP Edge Client on Access Policy Manager 10.x.....	18
Customizing an access policy to support BIG-IP Edge Client on Access Policy Manager 11.x.....	19
Chapter 3: Configuring a Connectivity Profile with Access Policy Manager	
Version 11.5 and later.....	21
About connectivity profiles.....	22
Creating a connectivity profile.....	22
Overview: Configuring APM for BIG-IP Edge Applications.....	22
Configuring security settings for iOS Edge Clients.....	22
Chapter 4: Overview: Access Policies for BIG-IP Edge Client.....	25
About access policy branches for BIG-IP Edge Client.....	26
Understanding basic access policy that supports BIG-IP Edge Client.....	26
Chapter 5: Configuring per-app VPN with APM and Edge Client.....	27
What is Per-App VPN?.....	28
About deploying MDM apps over VPNs.....	28
About access policies for Per-App VPN.....	29
Creating an access profile.....	29
Setting up Access Policy Manager for Per-App VPN.....	30
Configuring a virtual server for Per-App VPN.....	30
About managing devices.....	31
Creating a configuration profile for the managed device.....	31
Chapter 6: Additional Access Policy Manager Configuration Information.....	35
iOS clients using session variables.....	36
Access Policy Manager configuration tips.....	36

About starting the client from a URL scheme.....	37
About Edge Client Lite mode.....	38
Examples of starting a client from a URL.....	38
About defining a server from a URL.....	39
Examples of defining a server from a URL.....	40

Legal Notices

Publication Date

This document was published on October 13, 2014.

Publication Number

MAN-0393-06

Copyright

Copyright © 2014, F5 Networks, Inc. All rights reserved.

F5 Networks, Inc. (F5) believes the information it furnishes to be accurate and reliable. However, F5 assumes no responsibility for the use of this information, nor any infringement of patents or other rights of third parties which may result from its use. No license is granted by implication or otherwise under any patent, copyright, or other intellectual property right of F5 except as specifically described by applicable user licenses. F5 reserves the right to change specifications at any time without notice.

Trademarks

AAM, Access Policy Manager, Advanced Client Authentication, Advanced Firewall Manager, Advanced Routing, AFM, Application Acceleration Manager, Application Security Manager, APM, ARX, AskF5, ASM, BIG-IP, BIG-IQ, Cloud Extender, CloudFucious, Cloud Manager, Clustered Multiprocessing, CMP, COHESION, Data Manager, DevCentral, DevCentral [DESIGN], DNS Express, DSC, DSI, Edge Client, Edge Gateway, Edge Portal, ELEVATE, EM, Enterprise Manager, ENGAGE, F5, F5 [DESIGN], F5 Certified [DESIGN], F5 Networks, F5 SalesXchange [DESIGN], F5 Synthesis, f5 Synthesis, F5 Synthesis [DESIGN], F5 TechXchange [DESIGN], Fast Application Proxy, Fast Cache, FirePass, Global Traffic Manager, GTM, GUARDIAN, iApps, IBR, iCall, Intelligent Browser Referencing, Intelligent Compression, IPv6 Gateway, iControl, iHealth, iQuery, iRules, iRules OnDemand, iSession, L7 Rate Shaping, LC, Link Controller, LineRate, LineRate Systems [DESIGN], Local Traffic Manager, LROS, LTM, Message Security Manager, MobileSafe, MSM, OneConnect, Packet Velocity, PEM, Policy Enforcement Manager, Protocol Security Manager, PSM, Real Traffic Policy Builder, SalesXchange, ScaleN, SDAC (except in Japan), SDC, Signalling Delivery Controller, Solutions for an application world, Software Designed Applications Services, SSL Acceleration, StrongBox, SuperVIP, SYN Check, TCP Express, TDR, TechXchange, TMOS, TotALL, Traffic Management Operating System, Traffix (except Germany), Traffix [DESIGN] (except Germany), Transparent Data Reduction, UNITY, VAULT, vCMP, VE F5 [DESIGN], Versafe, Versafe [DESIGN], VIPRION, Virtual Clustered Multiprocessing, WebSafe, and ZoneRunner, are trademarks or service marks of F5 Networks, Inc., in the U.S. and other countries, and may not be used without F5's express written consent.

All other product and company names herein may be trademarks of their respective owners.

Patents

This product may be protected by one or more patents indicated at: <http://www.f5.com/about/guidelines-policies/patents>

Export Regulation Notice

This product may include cryptographic software. Under the Export Administration Act, the United States government may consider it a criminal offense to export this product from the United States.

RF Interference Warning

This is a Class A product. In a domestic environment this product may cause radio interference, in which case the user may be required to take adequate measures.

FCC Compliance

This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of FCC rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This unit generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference, in which case the user, at his own expense, will be required to take whatever measures may be required to correct the interference.

Any modifications to this device, unless expressly approved by the manufacturer, can void the user's authority to operate this equipment under part 15 of the FCC rules.

Canadian Regulatory Compliance

This Class A digital apparatus complies with Canadian ICES-003.

Standards Compliance

This product conforms to the IEC, European Union, ANSI/UL and Canadian CSA standards applicable to Information Technology products at the time of manufacture.

Acknowledgments

This product includes software developed by Bill Paul.

This product includes software developed by Jonathan Stone.

This product includes software developed by Manuel Bouyer.

This product includes software developed by Paul Richards.

This product includes software developed by the NetBSD Foundation, Inc. and its contributors.

This product includes software developed by the Politecnico di Torino, and its contributors.

This product includes software developed by the Swedish Institute of Computer Science and its contributors.

This product includes software developed by the University of California, Berkeley and its contributors.

This product includes software developed by the Computer Systems Engineering Group at the Lawrence Berkeley Laboratory.

This product includes software developed by Christopher G. Demetriou for the NetBSD Project.

This product includes software developed by Adam Glass.

This product includes software developed by Christian E. Hopps.

This product includes software developed by Dean Huxley.

This product includes software developed by John Kohl.

This product includes software developed by Paul Kranenburg.

This product includes software developed by Terrence R. Lambert.

This product includes software developed by Philip A. Nelson.

This product includes software developed by Herb Peyerl.

This product includes software developed by Jochen Pohl for the NetBSD Project.

This product includes software developed by Chris Provenzano.

This product includes software developed by Theo de Raadt.

This product includes software developed by David Muir Sharnoff.

This product includes software developed by SigmaSoft, Th. Lockert.

This product includes software developed for the NetBSD Project by Jason R. Thorpe.

This product includes software developed by Jason R. Thorpe for And Communications, <http://www.and.com>.

This product includes software developed for the NetBSD Project by Frank Van der Linden.

This product includes software developed for the NetBSD Project by John M. Vinopal.

This product includes software developed by Christos Zoulas.

This product includes software developed by the University of Vermont and State Agricultural College and Garrett A. Wollman.

This product includes software developed by Balazs Scheidler (bazsi@balabit.hu), which is protected under the GNU Public License.

This product includes software developed by Niels Mueller (nisse@lysator.liu.se), which is protected under the GNU Public License.

In the following statement, "This software" refers to the Mitsumi CD-ROM driver: This software was developed by Holger Veit and Brian Moore for use with 386BSD and similar operating systems. "Similar operating systems" includes mainly non-profit oriented systems for research and education, including but not restricted to NetBSD, FreeBSD, Mach (by CMU).

This product includes software developed by the Apache Group for use in the Apache HTTP server project (<http://www.apache.org/>).

This product includes software licensed from Richard H. Porter under the GNU Library General Public License (© 1998, Red Hat Software), www.gnu.org/copyleft/lgpl.html.

This product includes the standard version of Perl software licensed under the Perl Artistic License (© 1997, 1998 Tom Christiansen and Nathan Torkington). All rights reserved. You may find the most current standard version of Perl at <http://www.perl.com>.

This product includes software developed by Jared Minch.

This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (<http://www.openssl.org/>).

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com).

This product contains software based on oprofile, which is protected under the GNU Public License.

This product includes RRDtool software developed by Tobi Oetiker (<http://www.rrdtool.com/index.html>) and licensed under the GNU General Public License.

This product contains software licensed from Dr. Brian Gladman under the GNU General Public License (GPL).

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>).

This product includes Hypersonic SQL.

This product contains software developed by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and others.

This product includes software developed by the Internet Software Consortium.

This product includes software developed by Nominum, Inc. (<http://www.nominum.com>).

This product contains software developed by Broadcom Corporation, which is protected under the GNU Public License.

This product contains software developed by MaxMind LLC, and is protected under the GNU Lesser General Public License, as published by the Free Software Foundation.

This product includes Intel QuickAssist kernel module, library, and headers software licensed under the GNU General Public License (GPL).

This product includes software licensed from Gerald Combs (gerald@wireshark.org) under the GNU General Public License as published by the Free Software Foundation; either version 2 of the License, or any later version. Copyright ©1998 Gerald Combs.

This product includes software developed by Daniel Stenberg. Copyright ©1996 - 2012, Daniel Stenberg, (daniel@haxx.se). All rights reserved.

Permission to use, copy, modify, and distribute this software for any purpose with or without fee is hereby granted, provided that the above copyright notice and this permission notice appear in all copies.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Except as contained in this notice, the name of a copyright holder shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Software without prior written authorization of the copyright holder.

This product includes software developed by Thomas Williams and Colin Kelley. Copyright ©1986 - 1993, 1998, 2004, 2007

Permission to use, copy, and distribute this software and its documentation for any purpose with or without fee is hereby granted, provided that the above copyright notice appear in all copies and that both that copyright notice and this permission notice appear in supporting documentation. Permission to modify the software is granted, but not the right to distribute the complete modified source code. Modifications are to be distributed as patches to the released version. Permission to distribute binaries produced by compiling modified sources is granted, provided you

1. distribute the corresponding source modifications from the released version in the form of a patch file along with the binaries,
2. add special version identification to distinguish your version in addition to the base release version number,
3. provide your name and address as the primary contact for the support of your modified version, and
4. retain our contact information in regard to use of the base software.

Permission to distribute the released version of the source code along with corresponding source modifications in the form of a patch file is granted with same provisions 2 through 4 for binary distributions. This software is provided "as is" without express or implied warranty to the extent permitted by applicable law.

This product contains software developed by Google, Inc. Copyright ©2011 Google, Inc.

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

This product includes software developed by Jeremy Ashkenas and DocumentCloud, and distributed under the MIT license. Copyright © 2010-2013 Jeremy Ashkenas, DocumentCloud.

This product includes gson software, distributed under the Apache License version 2.0. Copyright © 2008-2011 Google Inc.

This product includes ec2-tools software, copyright © 2008, Amazon Web Services, and licensed under the Amazon Software License. A copy of the License is located at <http://aws.amazon.com/asl/>.

This product includes Boost libraries, which are distributed under the Boost license (http://www.boost.org/LICENSE_1_0.txt).

This product includes jxrlib software, copyright ©2009 Microsoft Corp. All rights reserved. Distributed under the new BSD license.

This product includes software developed by Computing Services at Carnegie Mellon University (<http://www.cmu.edu/computing/>).

This product includes libmagic software, copyright © Ian F. Darwin 1986, 1987, 1989, 1990, 1991, 1992, 1994, 1995. Software written by Ian F. Darwin and others; maintained 1994- Christos Zoulas.

This product contains OpenLDAP software, which is distributed under the OpenLDAP v2.8 license (BSD3-like).

This product includes node-uuid software, copyright © 2010-2012, Robert Kieffer, and distributed under the MIT license.

This product includes opensv software, which is distributed under the Apache 2.0 license.

This product includes owasp-jave-encoder software, copyright © 2014, Jeff Ichnowski, and distributed under the New BSD license.

Chapter

1

Overview: BIG-IP Edge Client for Mobile Devices

Topics:

- *What does BIG-IP Edge Client do for mobile devices?*

What does BIG-IP Edge Client do for mobile devices?

BIG-IP® Edge Client® 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 BIG-IP Edge Client, refer to the *BIG-IP® Edge Client® for iOS User Guide* on your device.

BIG-IP Edge Client features include:

- N-factor auth (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 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 3G 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.
- Support for SAML 2.0 features in BIG-IP® Access Policy Manager®.

About SAML support

The BIG-IP® Edge Client® application for mobile devices provide the following SAML support:

- SP-initiated access only (e.g. APM acting as the SP).
- Web Logon mode only.
- Single Log-Out (SLO) is supported only when the logout action is initiated from the client.

When you use Edge Client as a client performing SP-initiated access, Edge Client 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 Edge Client back to the SP with assertion. APM then accepts the assertion and establishes a VPN connection. You can then access backend resources through Edge Client.

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

The BIG-IP® Edge Client® application for mobile devices provides these authentication types:

Authentication type	Description
VPN On-Demand	Provides the following three options: <ul style="list-style-type: none"> • User name and password • Client certificate • Client certificate + user name and password (no runtime prompt)
Regular Logon	Provides the following three options: <ul style="list-style-type: none"> • User name and password • Client certificate • Client certificate + user name and password (no runtime prompt)

Authentication type	Description
Web Logon	Provides the following four options: <ul style="list-style-type: none"> User name and password User name/password + RSA + any other server-side checks Client certificate Client certificate + user name and password (no runtime prompt)

About establishing VPN connections

The BIG-IP® Edge Client® 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 BIG-IP Edge Client application, or implicitly through the VPN On-Demand functionality.

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

VPN On-Demand considerations:

- When VPN On-Demand initiates a connection, 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.
- VPN On-Demand supports only two authentication types. After you have imported the configuration profile, you can perform configurations to add additional credential authentication using the application.
- VPN On-Demand does not work if you enable 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.
- Log UDID information.

This example displays an access policy with a custom action of Device ID Check to check the device's UDID.

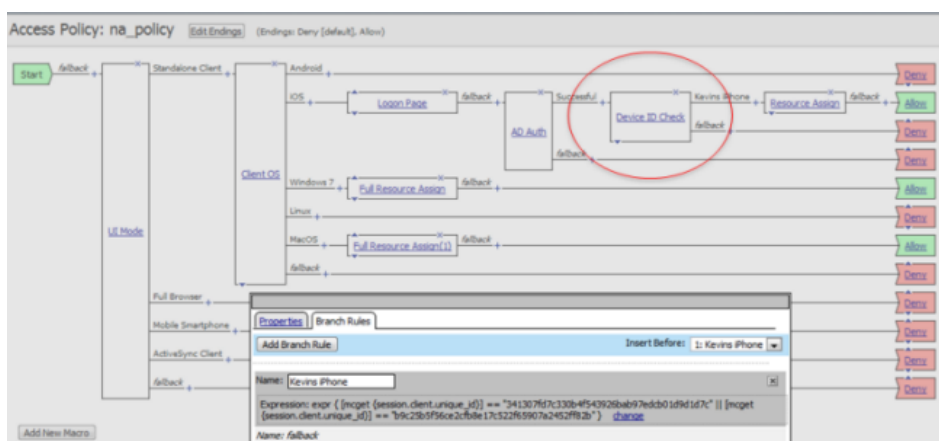


Figure 1: Example of a custom action for checking device's UDID

About automatically launching applications from mobile devices

You can configure BIG-IP® Edge Client® to launch an app with a registered URL scheme after a VPN connection is established.

Auto-launching applications from the BIG-IP Edge Client

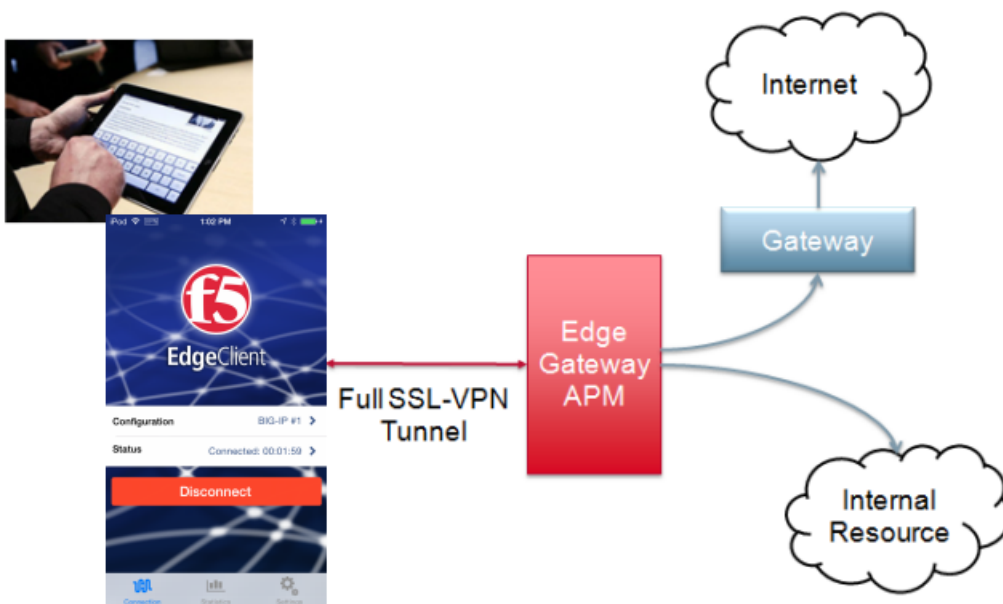
You can configure applications to automatically start on the BIG-IP Edge Client once a connection is initiated.

1. On the Main tab, click **Access Policy**.
2. Navigate to **Network Access**, and select the name of your network access from the list.
3. Select the **Launch Applications** tab.
4. In the **Application Path** field, type in your application path in the form of a URL scheme, for example, `skype://14082734800?call`.
5. From the **Operating System** selection field, select your operating system type.
6. Click **Finished**.

On the device itself, a pre-launch 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 a network access

You can force traffic through a tunnel on the BIG-IP® Edge Client®.

Note: Although you disable **Allow local subnet access** while enabling **Force all traffic through tunnel**, the client will still permit local subnet traffic to travel outside of the tunnel. This is a limitation of iOS and not with the BIG-IP Edge Client.

1. On the Main tab, click **Access Policy > Network Access**.
The Network Access List screen opens.
2. Click a name in the list to select a network access resource.
The network access properties screen opens.

3. To configure the network settings for the network access resource, click **Network Settings** on the menu bar.
4. For **Traffic Options**, enable **Force all traffic through tunnel**.

If you enable **Use split tunneling for traffic**, the client will use the proxy server if the destination matches the DNS address space or DNS suffixes. If you specify the destination using an IP address, the client will not use the proxy server, even if you include the proxy server in the split tunnel IP LAN address space.

If you enable **Force all traffic through tunnel**, the client will use the proxy server whether you specify the destination using an IP address or a host name.

5. For **Allow Local Subnet**, select the **Enable** check box.
6. For **Client Options**, enable **Client for Microsoft Networks**.
7. Click **Update**.

Chapter

2

Configuring Access Policy Manager for BIG-IP Edge Client


Topics:

- [*Access Policy Manager configuration for BIG-IP Edge Client for mobile devices*](#)

Access Policy Manager configuration for BIG-IP Edge Client for mobile devices

To configure BIG-IP® Edge Client® for mobile devices support on BIG-IP Access Policy Manager®, use these following configuration steps:

- Run the Network Access Setup Wizard.
- You can set up SSO and ACLs for your network access (optional). Refer to the *BIG-IP® Access Policy Manager® Configuration Guide* on the AskF5™ Knowledge Base for instructions.
- Customize an access policy to support BIG-IP Edge Client.

 **Important:** To resolve internal addresses with DNS, either the **Network Access DNS Address Space** or **DNS Default Domain Suffix** must be specified in the Network Access configuration. If neither field is configured, internal DNS addresses cannot be resolved. The DNS server must also be configured to resolve internal addresses with DNS.

Running the Network Access Setup Wizard


Configure Access Policy Manager® to provide users with full network access from their mobile 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 to BIG-IP® Edge Client®.
4. Click **Finished**.

You now have network access that supports BIG-IP Edge Client for mobile devices.

Customizing an access policy to support BIG-IP Edge Client on Access Policy Manager 10.x

Create an access policy that supports BIG-IP® Edge Client® for iOS.

 **Note:** This policy applies to Access Policy Manager® version 10.x systems.

1. On the Main tab, click **Access Policy > Access Profiles**.
The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the profile you want to configure to launch the visual policy editor.
The visual policy editor opens the access profile in a separate screen or tab.
3. Click the plus (+) sign that appears before the Logon Page action.
4. Under **Server Side Checks**, select **UI Mode**, and click **Add Item**.
5. Click **Save**.
The UI Mode action is added to the access policy, and several new branches appear.
6. On the Standalone Client branch of the UI Mode action, click the plus (+) sign.
7. Under **General Purpose**, select **Empty**, and click **Add Item**.
8. Click the Branch Rules tab.
9. Click **Add Branch Rule**.
10. Rename the new branch rule **Branch Rule n** to **iOS Edge Client**.
11. Next to **Expression: Empty**, click the **change** link.
12. Click the **Advanced** tab.

13. Type the following rule in the box: `expr { [mcget {session.client.platform}] == "iOS" }`
14. Click **Finished**, and then click **Save**.
15. Add the network access resource to the branch.
16. Click **Save**.

This access policy now supports BIG-IP Edge Client for iOS.

Customizing an access policy to support BIG-IP Edge Client on Access Policy Manager 11.x

Create an access policy that supports BIG-IP® Edge Client® for iOS.



Note: This policy applies to Access Policy Manager version 11.x systems.

1. On the Main tab, click **Access Policy > Access Profiles**.
The Access Profiles List screen opens.
2. In the Access Policy column, click the **Edit** link for the profile you want to configure to launch the visual policy editor.
The visual policy editor opens the access profile in a separate screen or tab.
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 Edge Client.
9. Click **Finished**, and then click **Save**.
10. Add the network access resource to the branch.
11. Click **Save**.

This access policy now supports BIG-IP Edge Client for iOS.

Chapter

3

Configuring a Connectivity Profile with Access Policy Manager Version 11.5 and later

Topics:

- [About connectivity profiles](#)
- [Overview: Configuring APM for BIG-IP Edge Applications](#)

About connectivity profiles

In BIG-IP® Access Policy Manager®, a connectivity profile is the profile that you select in a virtual server definition to define connectivity and client settings for a network access session.

The connectivity profile contains:

- Compression settings for network access connections and application tunnels
- Citrix client settings
- Virtual servers and DNS-location awareness settings for BIG-IP Edge Client® for Windows and Mac
- Password caching settings for BIG-IP Edge Client for Windows, Mac, and mobile clients
- Security settings, in addition to password caching, for mobile clients

A connectivity profile is also associated with client download packages that you can customize.

Creating a connectivity profile

You create a connectivity profile to configure client connections for a network access tunnel, application access tunnel, and clients.

1. On the Main tab, click **Access Policy > Secure Connectivity**.
A list of connectivity profiles displays.
2. Click **Add**.
The Create New Connectivity Profile popup screen opens and displays General Settings.
3. Type a **Profile Name** for the connectivity profile.
4. Select a **Parent Profile** from the list.
APM® provides a default profile, **connectivity**.
5. From the Compression Settings folder, click **Network Access** and make changes to the network access compression settings.
The settings specify available compression codecs for server-to-client connections.
The default settings are displayed in the right pane.
6. Click **OK**.
The popup screen closes, and the Connectivity Profile List displays.

The connectivity profile appears in the list.

To provide functionality with a connectivity profile, you must add the connectivity profile and an access profile to a virtual server.

Overview: Configuring APM for BIG-IP Edge Applications

A connectivity profile contains default settings for these mobile clients:

- BIG-IP® Edge Client® for Android
- BIG-IP Edge Portal® for Android
- BIG-IP Edge Client for iOS
- BIG-IP Edge Portal for iOS
- BIG-IP Edge Client for Windows Phone

The settings are security-related. They specify how to handle password caching (disabled by default in all cases), and device or PIN locking (enabled where supported). Customize the available settings to meet your requirements.

Configuring security settings for iOS Edge Clients

You must create a connectivity profile before you start this task.

A connectivity profile automatically contains default settings for BIG-IP® Edge Client® for iOS clients. You update the settings to change the way password caching and on demand disconnect are handled.

1. On the Main tab, click **Access Policy > Secure Connectivity**.
A list of connectivity profiles displays.
2. Select the connectivity profile that you want to update and click **Edit Profile**.
The Edit Connectivity Profile popup screen opens and displays General Settings.
3. From Mobile Client Settings in the left pane, select **iOS Edge Client**.
Settings for the iOS Edge Client display in the right pane.
4. If you want users to be able to save their passwords, select the **Allow Password Caching** check box.
5. For **Save Password Method**, specify how to use password caching:
 - To allow the user to save the encrypted password on the device without a time limit, select **disk**.
 - To specify that the user password is cached in the application on the user's device for a configurable period of time, select **memory**.

If you select **memory**, the **Password Cache Expiration (minutes)** field becomes available.

6. If the **Password Cache Expiration (minutes)** field displays, type the number of minutes you want the password to be cached in memory.
7. In the **On Demand Disconnect Timeout (minutes)** field, retain the default 2, or type a different number of minutes before VPN on demand times out.
8. Click **OK**.
The popup screen closes, and the Connectivity Profile List displays.

You have now customized the password caching and on demand disconnect settings for BIG-IP Edge Client for iOS clients.



Note: Password caching in memory mode is configured on the server. Because of this, VPN on Demand might stop working after being active for several hours. In this case, you must use manual connect on BIG-IP Edge Client to cache the password again. Password caching in disk mode saves the password in the VPN configuration. In this case, the password is saved every time on both manual connect and VPN on Demand.

To provide functionality with a connectivity profile, you must add the connectivity profile and an access profile to a virtual server.

Chapter

4

Overview: Access Policies for BIG-IP Edge Client

Topics:

- [About access policy branches for BIG-IP Edge Client](#)

About access policy branches for BIG-IP Edge Client

You can configure separate access policy branches for BIG-IP® Edge Client®.

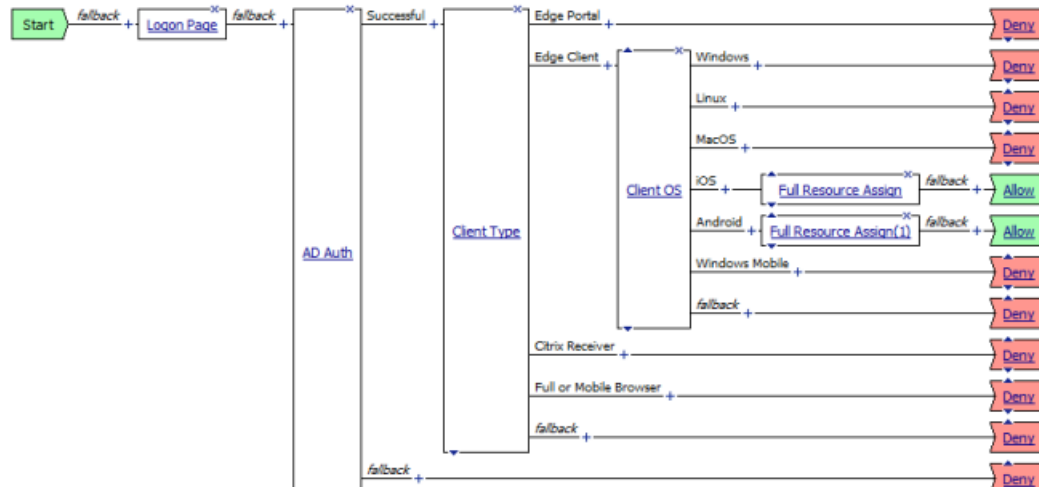
BIG-IP Edge Client does not support client-side checks; however, you can configure an access policy that provides network access for iOS clients by using any of these methods:

- Create an access policy using **Client-Side Check 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.
- Create a specific branch for iOS clients. Use an empty action and empty session variables to identify the client. Add authentication and assign a network access resource for iOS clients to this branch.

Understanding basic access policy that supports BIG-IP Edge Client

You can configure an access policy branch to direct mobile device users to BIG-IP® Edge Client®, and direct non-mobile device users to a fallback branch.

This example displays a simple access policy.



Chapter

5

Configuring per-app VPN with APM and Edge Client

Topics:


- *What is Per-App VPN?*
- *About deploying MDM apps over VPNs*
- *About access policies for Per-App VPN*
- *Setting up Access Policy Manager for Per-App VPN*
- *About managing devices*

What is Per-App VPN?

With iOS 7, Apple enhanced their VPN framework to support application level layer-4 tunneling. Apps that are managed by a Mobile Device Manager (MDM) can now be configured to automatically connect to a VPN when they are started. 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 four configuration components.

- A device under MDM management.
- A managed app installed on the device, or Mobile Safari.
- The iOS Edge Client installed on the managed device.
- A per-app VPN profile, and a related Edge Client Configuration (VPN). This is configured with an MDM command that associates the App with an Edge Client Configuration.

 **Important:** The managed App and the MDM profile must be deployed with an MDM solution, except in the case of Mobile Safari. The Edge Client and Edge Client 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 specific apps only. Specifically, it allows applications to use one and only one Edge Client Configuration (or VPN connection).

 **Important:** If the Edge Client Configuration is not connected when the app starts, all traffic from the app is blocked.

In practice, some applications may be associated with one Edge Client Configuration, and other applications may be associated with other Edge Client Configurations.

 **Important:** Once an app is associated with an Edge Client Configuration by the MDM, it must use that VPN only.

On iOS 7, only one Edge client Configuration can be active at one time. In this example, App 1 or App 2 can be active at different times, because they use different VPN configurations. If App 2 is made active, and starts the VPN2 connection, then App 1 cannot use a VPN connection, as it is associated with VPN 1. Since split tunneling is not allowed, App1 has no access to any network services as long as App2 is active and using VPN2.

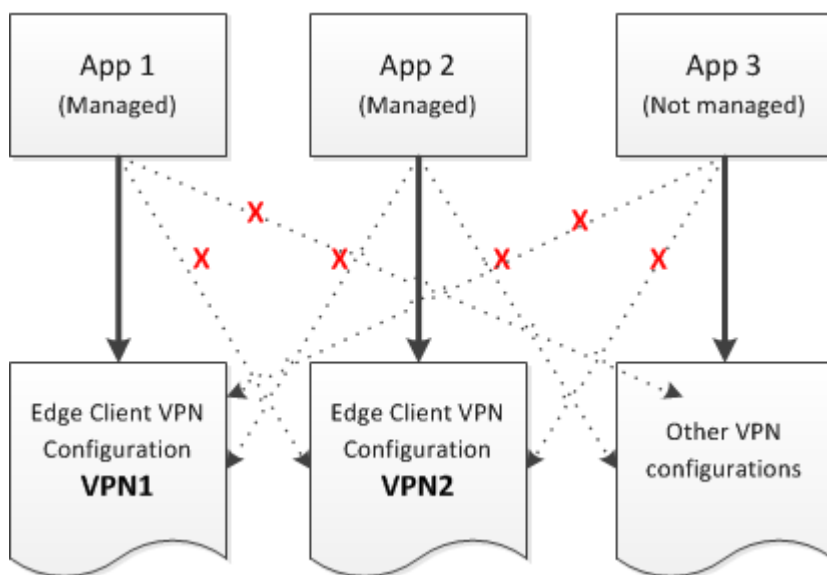



Figure 2: Apps associated with different VPN configurations

 **Note:** On iOS 8 and above, you can only activate one L3 VPN configuration at a time. However, multiple Per-App VPN can be active and connected simultaneously.

About access policies for Per-App VPN

For Per-App VPN, an access policy requires a specific configuration. In particular, the Per-App VPN process cannot allow prompts or request information during logon. Therefore the access policy must be configured to log the user on to the connection without any user interaction.

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.

Configure the access policy to include a client certificate check.

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 Policy > Access Profiles**.
The Access Profiles List screen opens.
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. 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 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.

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 SOCKS 5 proxy tunnels, and do not require Network Access resources in the Access Policy. Configure these items on the Access Policy Manager.

- The virtual server must be configured with several critical settings, including an access profile and the **Application Tunnels (Java & Per-App VPN)** setting.
- If there is routing required behind the BIG-IP device, the SNAT Automap should be enabled.
- The SOCKS proxy has too much access by default, and you should configure ACLs to limit access to only the required URLs and ports.
- 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.

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 SOCKS proxy method 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. From the **Source Address Translation** list, select **Auto Map**.
The BIG-IP® system uses all of the self IP addresses as the translation addresses for the pool.
5. From the Access Policy area, select the **VDI & Java Support** check box.
6. In the Access Policy area, from the **Access Profile** list, select the access profile.
7. From the **Connectivity Profile** list, select the connectivity profile.
8. Click **Update** to save the changes.

The virtual server is configured for Per-App VPN.

About managing devices

With an MDM, you manage devices by enrolling them. Refer to your MDM documentation to enroll devices.

-  **Important:** A user must enroll the device with the MDM in order for you to deploy managed apps to the client. In addition, the iOS Edge Client must be installed on the device. The Edge Client can be installed either by the user, or deployed with the MDM solution.

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, the iOS Edge Client 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.

1. Create a configuration profile for the device.
Configuration profiles are described at the [Apple Configuration Profile Reference](#).
2. In the configuration profile, in the VendorConfig section, specify a PerAppVpn key with the syntax `<key>PerAppVpn</key><string>true</string>`.
3. Specify an app by sending the `InstallApplication` command or the `Settings` command.
These settings can be configured only for apps that are installed and managed by the MDM.
4. Specify which managed apps use Per-App VPN by sending the `InstallApplication` or `Settings` command.
Per-App VPN can be specified only for MDM-managed apps. The only exception is Mobile Safari. For Mobile Safari, the admin can specify domains in the profile that start the Per-App VPN connection.

Configure Access Policy Manager to provide the necessary support for Per-App VPN features.

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 1: Per-App VPN specific keys

Key	Type	Description
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	Boolean	<p>If true, the Per-App VPN connection starts automatically when apps linked to this Per-App VPN service initiate network communication.</p> <p>If false, the Per-App VPN connection must be started manually by the user before apps linked to this Per-App VPN service can initiate network communication.</p> <p>If this key is not present, the value of the <code>OnDemandEnabled</code> key is used to determine the status of Per-App VPN On Demand.</p>

Key	Type	Description
SafariDomains	Array	<p>This optional 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.</p> <p>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:</p> <ul style="list-style-type: none"> • 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 "apple.com" matches "www.apple.com", but not "foo.badapple.com". • Domain strings with only one label must match the entire host string. For example, a domain of "com" matches "com", not "www.apple.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, Payload UUID, 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.siterequest.mdm.perapp.vpn.vpn</string>
      <key>PayloadOrganization</key>
      <string></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>
  </array>
</dict>
</plist>
```



```

</dict>
<key>UserDefinedName</key>
<string>Per-App VPN Test</string>
<key>VPN</key>
<dict>
  <key>AuthName</key>
  <string>testuser</string>
  <key>AuthPassword</key>
  <string>testpassword</string>
  <key>AuthenticationMethod</key>
  <string>Certificate</string>
  <key>PayloadCertificateUUID</key>
  <string>C9BF4927-E819-4521-88DE-2AEB6E1DC3D8</string>
  <key>RemoteAddress</key>
  <string>vpn.siterequest.com</string>
  <key>OnDemandMatchAppEnabled</key>
  <true/>
</dict>
<key>VPNSubType</key>
<string>com.f5.F5-Edge-Client.vpnplugin</string>
<key>VPNType</key>
<string>VPN</string>
<key>VendorConfig</key>
<dict>
  <key>PerAppVpn</key>
  <string>true</string>
</dict>
<key>SafariDomains</key>
<array>
  <string>safaridomain1.com</string>
  <string>safaridomain2.com</string>
</array>
<key>VPNUUID</key>
<string>9F658A35-2B0F-4D5E-872D-61A9130FE882</string>
</dict>
<dict>
  <key>Password</key>
  <string>123456</string>
  <key>PayloadCertificateFileName</key>
  <string>identity.pl2</string>
  <key>PayloadContent</key>
  <data>
    MIIL2QIBAzCCC58GCSqGSIb3DQEHAaCCC5AEgguMMIILiDCCBj8G
    .....<truncated for example>.....
    hxd6YPi7JKB/24dSls9gKO/DHVoECHap2RUyKvQTAgIIAA==
  </data>
  <key>PayloadDescription</key>
  <string>Provides device authentication (certificate or identity).</
string>
  <key>PayloadDisplayName</key>
  <string>identity.pl2</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>
```

Chapter

6

Additional Access Policy Manager Configuration Information

Topics:

- [*iOS clients using session variables*](#)
- [*Access Policy Manager configuration tips*](#)
- [*About starting the client from a URL scheme*](#)
- [*About defining a server from a URL*](#)

iOS clients using session variables

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

Session variable	Description
<i>session.client.type</i>	Indicates the client type, such as Standalone.
<i>session.client.platform</i>	Indicates the platform type, such as iOS.
<i>session.client.agent</i>	Indicates the browser, device type, and operating system version of the client, as well as the version of BIG-IP Edge Client.
<i>session.client.model</i>	Indicates the model number of the mobile device. Sample string: %session.client.model%= 'Nexus One'
<i>session.client.platform_version</i>	Indicates the platform and version of the mobile device. Sample string: %session.client.platform_version%= '2.3.3'
<i>session.client.unique_id</i>	Indicates the unique ID of the mobile device. Sample string: %session.client.unique_id%= '8ccaf965e51e3077'
<i>session.client.jailbreak</i>	Indicates the jailbreak status of the device. Sample string: %session.client.jailbreak%= '0', where 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.

Access Policy Manager configuration tips

The following table provides tips for setting up BIG-IP® Edge Client® for mobile devices.

Feature	Information
VPN On-Demand (iOS only)	A connection cannot be established if the server has an invalid certificate. To work around this issue, manually import the invalid certificate onto the device.
Client endpoint checks	Client end-point checks are not currently supported.
Password caching policy	<ul style="list-style-type: none"> Under Client Policy, if Enforce session settings is not enabled, clients can save their encrypted password to disk, regardless of what settings are configured under Session Settings. Under Password Caching Options if you set Cache password within application for for a specific amount of time, after a successful logon the submitted credentials are cached until one of the following occurs: <ul style="list-style-type: none"> The specified credential cache duration expires. The server address of the configuration within the application changes. The user name of the configuration within the application changes. The BIG-IP Edge Client user switches between configurations and makes a new connection. The configuration is deleted and a new one is created. On the mobile device, even if a user clicks Disconnect, then terminates the application or restarts the device, cached credentials are not cleared until the specified cache time.
Client certificates	Client certificate authentication is supported, either with a certificate alone or with a certificate secured with a user name and password.

Feature	Information
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 BIG-IP® Edge Client® 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 (app). 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.

```
f5edgeclient://{start|stop}?[parameter1=value1&parameter2=value2...]
```

You can start an alternate "light" client with no client branding, using the following parameters.

```
f5edgeclient-lite://{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 the Edge Client 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. 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.

sid

A parameter used to specify the session ID with which to start the connection. When the parameter sid is provided, the username and password parameters are ignored, and no additional authentication occurs.

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*.

About Edge Client Lite mode

You can use a URL parameter to start **Edge Client Lite** mode. Edge Client Lite removes branding and presents a plain black screen for the Edge Client. In Edge Client Lite mode, the client has certain features and restrictions.

f5edgeclient-lite://

Provide a URL parameter that begins with `f5edgeclient-lite://` to start a connection in Edge Client Lite mode.

Plain black screen

When the app starts from an `f5edgeclient-lite://` URL, a black screen appears for the Edge Client configuration, hiding most controls from the user.

Text displayed

The Edge Client Lite displays status messages while the connection starts, and a **Cancel** button.

Authentication

Authentication or confirmation prompts appear over the black screen.

Standard Edge Client interface available after app switch

If the user switches away from the Edge Client, and then returns to it, the standard Edge Client interface appears, not blacked out. The Edge Client Lite user interface appears at initial logon only.

x-callback-url

The `x-callback-url` allows you to send one or more messages back to another app that has a registered URL scheme. The available messages are `x-success`, `x-cancel`, and `x-error`.

Standard logon mode only

Edge Client Lite does not work with web logon mode.

Examples of starting a client from a URL

The following examples illustrate how to start BIG-IP® Edge Client® and Edge Client Lite connections for users from a URL.

Connecting to an existing configuration called MYVPN:

```
f5edgeclient://start?name=MYVPN
```

Connecting to an existing configuration called MYVPN with Edge Client Lite:

```
f5edgeclient-lite://start?name=MYVPN
```

Connecting to an existing configuration called MYVPN with Edge Client Lite, from an app registered with the URL scheme `refapp://`, and specifying `x-callback-url` parameters:

```
f5edgeclient-lite://start?name=MYVPN&x-success=refapp%3A%2F%2Fsuccess&x-cancel=refapp%3A%2F%2Fcancel&x-error=refapp%3A%2F%2Ferror
```

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

```
f5edgeclient://start?name=MYVPN&server=myvpn.siterequest.com
```

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

```
f5edgeclient://start?server=myvpn.siterequest.com
```

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

```
f5edgeclient://start?server=myvpn.siterequest.com&logon_mode=web
```

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

```
f5edgeclient://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:

```
f5edgeclient://start?name=MYVPN&postlaunch_url=jump%3A%2F%2F%3Fhost%3D10.10.1.10%26username%3Dsmith
```

Stopping a connection:

```
f5edgeclient://stop
```

About defining a server from a URL

You can add BIG-IP® server definitions to Edge Client® from a URL. You can provide these URLs to users, so they can start and save VPN connections without having to manually start the application.

Use the following URL and parameters to create a server:

```
f5edgeclient://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.



Important: When username is specified, Web Logon is automatically disabled.

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.  Important: When password is specified, Web Logon is automatically disabled.
certcn	Certificate common name. Matches the Common Name of a valid certificate pre-installed on the device.  Important: When certcn is specified, Web Logon is automatically disabled.
domain_always	An optional, comma-separated list of match pattern(s) for the Always Connect domain list, for iOS devices only.
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.
logon_mode	An optional parameter that specifies whether the logon mode is the standard logon (<i>native</i>) or web logon (<i>web</i>). The default logon mode is <i>native</i> .

Examples of defining a server from a URL

The following examples illustrate how to define servers for BIG-IP® Edge Client® connections from a URL.

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

```
f5edgeclient://create?server=edgeportal.siterequest.com
```

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

In this scenario, both name and server are specified, and username and certcn are absent, so web logon is assumed.

```
f5edgeclient://create?name=EdgePortal&server=edgeportal.siterequest.com
```

Create the same server with a user name, password, and certificate:

```
f5edgeclient://create?name=EdgePortal&server=edgeportal.siterequest.com&username=edgeportal&password=androiddemo&certcn=clientcert-cert.siterequest.com
```

Create the same server with a user name and certificate:

```
f5edgeclient://create?name=EdgePortal&server=edgeportal.siterequest.com&username=edgeportal&certcn=clientcert-cert.siterequest.com
```

Create the same server with a certificate:


```
f5edgeclient://create?name=EdgePortal&server=
edgeportal.siterequest.com&certcn= clientcert-
cert.siterequest.com
```

