

BIG-IP® Access Policy Manager®: Visual Policy Editor

Version 11.5.2



Table of Contents

Legal Notices.....	7
Acknowledgments.....	9
 Chapter 1: Visual Policy Editor.....	 13
About the visual policy editor.....	14
Visual policy editor conventions.....	14
About actions on the add item screen.....	15
About macrocalls on the add item screen.....	16
About macros and macrocalls.....	17
Additional resources and documentation for BIG-IP Access Policy Manager.....	18
 Chapter 2: Defining Access Policy Items.....	 21
About access policy item configuration.....	22
Adding a blank access policy item to an access policy.....	23
Adding an access policy item with preconfigured branch rules.....	23
Adding an access policy item with configurable properties.....	25
Adding an access policy assignment item.....	26
Adding an access policy mapping item.....	27
 Chapter 3: Access Policy Item Reference.....	 29
About logon items.....	30
About the External Logon page.....	30
About HTTP 401 Response	31
About HTTP 407 Response.....	32
About logon page actions.....	32
About the virtual keyboard.....	34
About the VMware View logon page action.....	34
About assignment items.....	35
About ACL Assign	36
About AD Group Resource Assign	36
About Advanced Resource Assign	36
About BWC Policy	37
About Citrix Smart Access	37
About Dynamic ACL	37
About LDAP Group Resource Assign	38
About Pool Assign	38
About resource assignment.....	39
About Route Domain and SNAT Selection	39
About SSO Credential Mapping	40
About SWG Scheme Assign	40

About Variable Assign	40
About Webtop and Links	41
About endpoint security client-side items.....	42
About client-side action requirements and alternatives.....	42
About the Anti-spyware action.....	42
About the Antivirus action.....	43
About the Firewall action.....	44
About Hard Disk Encryption	45
About Linux File	45
About Linux Process	46
About Mac File	46
About Mac Process	47
About Machine Cert Auth	47
About Machine Info	49
About Patch Management	50
About Peer-to-Peer	51
About Windows Cache and Session Control	52
About the Windows File action.....	53
About Windows Health Agent	53
About Windows Info	54
About Windows Process	54
About Windows Protected Workspace	55
About Windows Registry	56
About 32-bit registry keys on a 64-bit Windows client.....	57
About endpoint security (server-side) access policy items.....	57
About Client for MS Exchange	57
About Client OS	58
About Client Type	59
About Client-Side Capability	60
About the Date Time action.....	61
About IP Geolocation Match	61
About IP Reputation	62
About IP Subnet Match	62
About Jailbroken or Rooted Device Detection	62
About Landing URI	62
About the License action.....	63
About general purpose items.....	63
About the Decision Box action.....	64
About the Email action.....	65
About the Empty action.....	66
About iRule Event	66
About Local Database	66
About the Logging action.....	67
About the Message Box action.....	67
About authentication items.....	68

About AD Auth	68
About AD Query	69
About Client Cert Inspection	70
About CRLDP Auth	70
About HTTP Auth	70
About Kerberos Auth	71
About LDAP Query	71
About LocalDB Auth	72
About NTLM Auth Result	72
About OAM authentication.....	72
About OCSP Auth	73
About On-Demand Cert Auth	73
About OTP Generate	73
About OTP Verify	74
About SAML Auth	74
About RADIUS Acct	74
About RADIUS Auth	74
About RSA SecurID	75
About TACACS+ Acct	75
About TACACS+ Auth.....	75
About Transparent Identity Import.....	76
 Chapter 4: Session Variables.....	77
About session variables.....	78
About session variable names.....	78
Session variables reference.....	79
 Chapter 5: Tcl Usage.....	85
About Tcl usage in APM.....	86
Tcl syntax notes.....	86
Tcl examples.....	87

Legal Notices

Publication Date

This document was published on January 28, 2015.

Publication Number

MAN-0507-02

Copyright

Copyright © 2013-2015, 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, APM, Application Acceleration Manager, Application Security Manager, 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, Intelligent Browser Referencing, Intelligent Compression, IPv6 Gateway, iControl, iHealth, iQuery, iRules, iRules OnDemand, iSession, L7 Rate Shaping, LC, Link Controller, Local Traffic Manager, LTM, LineRate, LineRate Systems [DESIGN], LROS, LTM, Message Security Manager, MSM, OneConnect, Packet Velocity, PEM, Policy Enforcement Manager, Protocol Security Manager, PSM, Real Traffic Policy Builder, SalesXchange, ScaleN, Signalling Delivery Controller, SDC, SSL Acceleration, software designed applications services, SDAC (except in Japan), StrongBox, SuperVIP, SYN Check, TCP Express, TDR, TechXchange, TMOS, TotALL, Traffic Management Operating System, Traffix Systems, Traffix Systems (DESIGN), 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.

Acknowledgments

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

Chapter 1

Visual Policy Editor

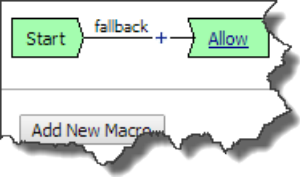
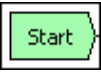


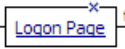


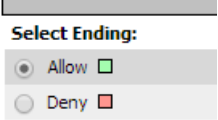

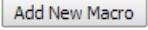
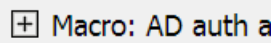
- *About the visual policy editor*
- *Visual policy editor conventions*
- *About actions on the add item screen*
- *About macrocalls on the add item screen*
- *About macros and macrocalls*
- *Additional resources and documentation for BIG-IP Access Policy Manager*

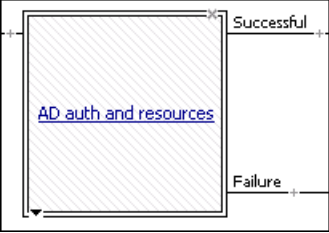
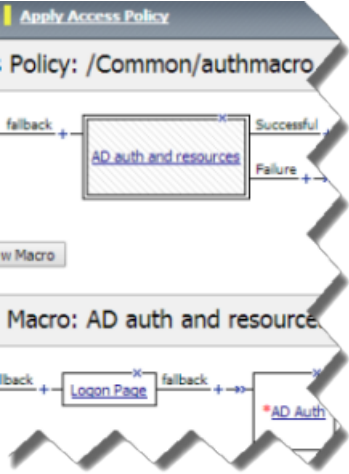
About the visual policy editor

The visual policy editor is a screen on which to configure an access policy using visual elements.

Visual policy editor conventions

This table provides a visual dictionary for the visual policy editor.

Visual element	Element type	Description
	Initial access policy	When an access profile is created, usually an initial access policy is also created.
	Start	Every access profile contains a start.
	Branch	A branch connects an action to another action or to an ending.
	Add an action	Clicking this icon causes a screen to open with available actions for selection.
	Action	Clicking the name of an action, such as Logon Page , opens a screen with properties and rules for the action. Clicking the x deletes the action from the access policy.
	Action that requires some configuration	The red asterisk indicates that some properties must be configured. Clicking the name opens a screen with properties for the action.
	Ending	Each branch has an ending: Allow or Deny .
	Configure ending	Clicking the name of an ending opens a popup screen.
	Add a macro for use in the access policy	Opens a screen for macro template selection. After addition, the macro is available for configuration and for use as an action item.
	Macro added for use	Added macros display under the access policy. Clicking the plus (+) sign expands the macro for configuration of the actions in it.
		

Visual element	Element type	Description
	Macrocall in an access policy	Clicking the macrocall name expands the macro in the area below the access policy.
	Apply Access Policy	Clicking it commits changes. The visual policy editor displays this link when any changes remain uncommitted.

About actions on the add item screen

The actions that are available on any given tab of the add item screen depend on the access profile type, such as LTM-APM (for web access) or SSL-VPN (for remote access), and so on. Only actions that are appropriate for the access profile type will display.

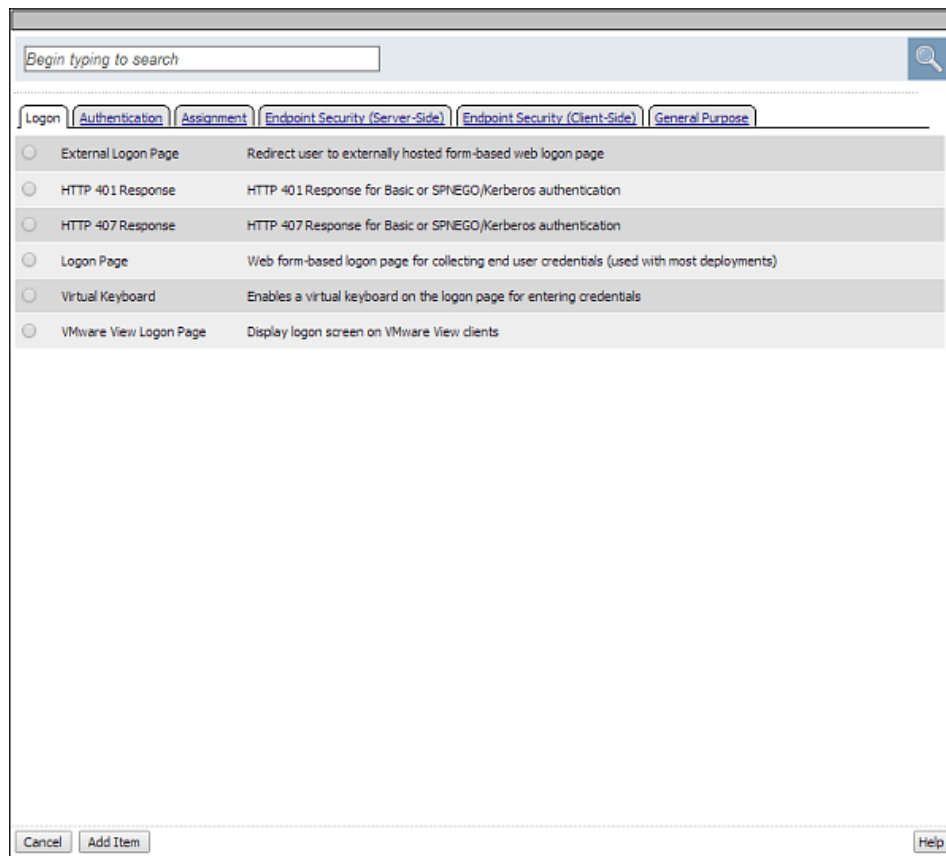


Figure 1: Add action item screen

About macrocalls on the add item screen

The Macrocalls tab displays only when at least one macro has been added for use in the access policy.

Note: *Macrocalls can be added to any access policy. Macrocalls cannot be shared across access policies.*

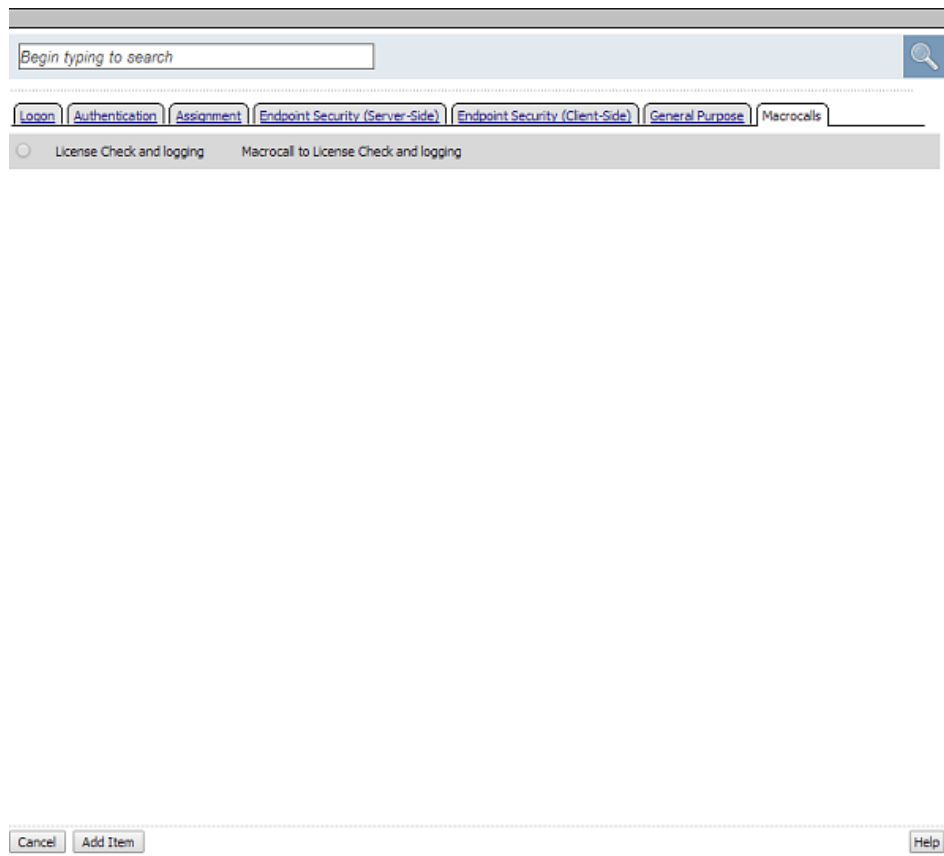


Figure 2: Macrocalls tab on the add item screen

About macros and macrocalls

A *macro* is a collection of access policy actions that provide common access policy functions. For example, AD auth and resources is a preconfigured macro template. It supplies a logon page, an Active Directory authentication action, and a resource assignment action. The properties and rules for the actions are configurable.

After a macro is configured, it can be placed into the access policy by adding a macrocall. A *macrocall* is an action that performs the functions defined in a macro.

A macro contains actions and terminals and can include macrocalls.

Access policy actions

Any available action or series of actions.

Macrocalls

Calls to other macros (nested macros).

Terminals

An endpoint in a macro. Default terminals are **Successful** and **Failure**. Terminals are configurable and can be added and deleted.

Terminals defined in the macro display as the branches that follow the macrocall after it has been added to the access policy.

Additional resources and documentation for BIG-IP Access Policy Manager

You can access all of the following BIG-IP® system documentation from the AskF5™ Knowledge Base located at <http://support.f5.com/>.

Document	Description
<i>BIG-IP® Access Policy Manager®: Secure Web Gateway Implementations</i>	This guide contains information to help an administrator configure Secure Web Gateway (SWG) explicit or transparent forward proxy and apply URL categorization and filtering to Internet traffic from your enterprise.
<i>BIG-IP® Access Policy Manager®: Third-Party Integration Implementations</i>	This guide contains information about integrating third-party products with Access Policy Manager (APM®). It includes implementations for integration with VMware Horizon View, Oracle Access Manager, Citrix Web Interface site, and so on.
<i>BIG-IP® Access Policy Manager®: Authentication and Single-Sign On</i>	This guide contains information to help an administrator configure APM for single sign-on and for various types of authentication, such as AAA server, SAML, certificate inspection, local user database, and so on.
<i>BIG-IP® Access Policy Manager®: Visual Policy Editor</i>	This guide contains information about how to use the visual policy editor to configure access policies.
<i>BIG-IP® Access Policy Manager®: Implementations</i>	This guide contains implementations for synchronizing access policies across BIG-IP systems, hosting content on a BIG-IP system, maintaining OPSWAT libraries, configuring dynamic ACLs, web access management, and configuring an access policy for routing.
<i>BIG-IP® Access Policy Manager®: Portal Access</i>	This guide contains information about how to configure APM portal access. In portal access, APM communicates with back-end servers, rewrites links in application web pages, and directs additional requests from clients back to APM.
<i>BIG-IP® Access Policy Manager®: Edge Client and Application Configuration</i>	<p>This guide contains information for an administrator to configure the BIG-IP system for these clients:</p> <ul style="list-style-type: none"> • BIG-IP® Edge Client® for Windows • BIG-IP Edge Client for Mac • BIG-IP Edge Client for Linux • BIG-IP Edge Command-Line Client for Linux <p>It also includes information about how to configure or obtain client packages and install them, as well as configuration details of system security settings on the BIG-IP system for these applications:</p> <ul style="list-style-type: none"> • BIG-IP Edge Client for iOS • BIG-IP Edge Client for Android • BIG-IP® Edge Portal® for iOS • BIG-IP Edge Portal for Android
<i>BIG-IP® Access Policy Manager®: Application Access</i>	This guide contains information for an administrator to configure application tunnels for secure, application-level TCP/IP connections from the client to the network.

Document	Description
<i>BIG-IP® Access Policy Manager®: Network Access</i>	This guide contains information for an administrator to configure APM network access to provide secure access to corporate applications and data using a standard web browser.
<i>BIG-IP® Access Policy Manager®: Customization</i>	This guide provides information about using the APM customization tool to provide users with a personalized experience for access policy screens, and errors. An administrator can apply your organization's brand images and colors, change messages and errors for local languages, and change the layout of user pages and screens.
Release notes	Release notes contain information about the current software release, including a list of associated documentation, a summary of new features, enhancements, fixes, known issues, and available workarounds.
Solutions and Tech Notes	Solutions are responses and resolutions to known issues. Tech Notes provide additional configuration instructions and how-to information.

Chapter

2

Defining Access Policy Items

- *About access policy item configuration*
- *Adding a blank access policy item to an access policy*
- *Adding an access policy item with preconfigured branch rules*
- *Adding an access policy item with configurable properties*
- *Adding an access policy assignment item*
- *Adding an access policy mapping item*

About access policy item configuration

An access policy item is a small action, or rule, that serves a specific purpose in an access policy. Access policy items are all added to the access policy in the same way, but in most cases, each access policy item must be configured individually. In Access Policy Manager®, an access policy item is one of five types.

Item type	Configuration details	Examples
Blank item	This type of access policy item has no explicit configuration on the configuration page, and can be configured to check a wide range of conditions with Expression windows.	<ul style="list-style-type: none"> General Purpose: Empty action Endpoint Security (Client-Side): Machine Info
Preconfigured branch rule item	This type of access policy item has no explicit configuration on the configuration page, and a preconfigured set of rules on the Branch Rules page.	<ul style="list-style-type: none"> Endpoint Security (Server-Side): IP Reputation Endpoint Security (Client-Side): Windows Info
Properties page configuration item	This type of access policy has all standard configuration options on the configuration page, to check the required information, prompt for information, or another action.	<ul style="list-style-type: none"> General Purpose: Logon Page action Endpoint Security (Client-Side): Antivirus
Assignment item	An assignment action allows configuration on the configuration page, and contains a list of available resources of a certain type, and allows you to select one or multiple resources to assign. Some resource assignment action, such as Webtop and Links assign, allow you to assign multiple items of different types. Advanced Resource Assign is a special case that allows you to select and assign multiple resources of different types at once.	<ul style="list-style-type: none"> Assignment: Pool Assign Assignment: Webtop and Links Assign
Mapping assignment item	A mapping assignment action allows you to assign one variable or resource to the value of another variable or resource. This kind of assign action includes the assignment of resources or variables on a separate page, linked from the main screen.	<ul style="list-style-type: none"> Assignment: AD Group Resource Assign Assignment: Variable Assign

Adding a blank access policy item to an access policy

Before you start this task, configure an access profile.

Configure a blank item to configure one of several actions that has no explicit configuration defined.

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.
The visual policy editor opens the access policy in a separate screen.
3. Click the (+) icon anywhere in the access policy to add a new action item.
A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.

4. Select a blank action:

Option	Description
Endpoint Security (Client-Side) > Machine Info	Collects machine info, and checks it against established values.
General Purpose > Empty	An empty action that you can configure with any allowed checks.

A properties screen opens.

5. Click the Branch Rules tab.
The Branch Rules screen opens.
6. Click the **Add Branch Rule** button.
New **Name** and **Expression** settings display.
7. Click the **change** link in the Expression section.
A popup screen opens.
8. Click **Add Expression**.
New properties display.
9. For each expression you add, select an agent from the **Agent Sel.** list, a condition from the **Condition** list, and configure any details.
See the reference information for each action for more details.
10. Click **Add Expression** to add the expression to the list.
11. Add more expressions to the check as required. You can add expressions as either **AND** or **OR** conditions.
12. Click **Finished**.
The popup screen closes.
13. Click **Save**.
The properties screen closes and the visual policy editor displays.

The access policy is configured with the empty action you have configured.

Click the **Apply Access Policy** link to apply and activate your changes to this access policy.

Adding an access policy item with preconfigured branch rules

Before you start this task, configure an access profile.

Configure an access policy with preconfigured branch rules to add preconfigured settings and branches to an access policy.

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.
The visual policy editor opens the access policy in a separate screen.
3. Click the (+) icon anywhere in the access policy to add a new action item.
A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. Select an action with preconfigured branch rules, and click **Add Item**:

Option	Description
Endpoint Security (Server-Side) > Client for MS Exchange	Checks that the system is a client for Microsoft Exchange.
Endpoint Security (Server-Side) > Client OS	Provides branches based on the result of an operating system check on the client.
Endpoint Security (Server-Side) > Client Type	Provides branches based on the result of a client type check.
Endpoint Security (Server-Side) > Client-Side Capability	Checks whether the client can run client side checks and provides positive and fallback branches.
Endpoint Security (Server-Side) > Date Time	Provides branches based on a certain date or time.
Endpoint Security (Server-Side) > IP Geolocation Match	Provides branches based on a specific geographic origin for the client.
Endpoint Security (Server-Side) > IP Reputation	Checks the client IP against an IP reputation database.
Endpoint Security (Server-Side) > Jailbroken or Rooted Device Detection	Provides branches based on whether the device appears to be jailbroken or rooted.
Endpoint Security (Server-Side) > Landing URI	Provides branches based on a specific landing URI.
Endpoint Security (Server-Side) > License	Provides branches based on the available global APM licenses.
Endpoint Security (Client-Side) > Windows Info	Provides branches based on specific Windows information, such as operating system type and patch level.

A properties screen opens.

5. Click the Branch Rules tab.
The Branch Rules screen opens.
6. View the preconfigured branch rules.
You can make changes to the branch rules, or close the item.
7. Click **Save**.
The properties screen closes and the visual policy editor displays.

The access policy is saved with the action you have configured.

Click the **Apply Access Policy** link to apply and activate your changes to this access policy.

Adding an access policy item with configurable properties

Before you start this task, configure an access profile.

Configure an access policy with configurable properties to check for specific items or policies.

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.
The visual policy editor opens the access policy in a separate screen.
3. Click the (+) icon anywhere in the access policy to add a new action item.
A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
4. Select an action with configurable properties, then click **Add Item**:

Option	Description
Logon > External Logon Page	Presents an external logon page for the client.
Logon > HTTP 401 Response	Provides a custom HTTP 401 logon page.
Logon > HTTP 407 Response	Provides a custom HTTP 407 logon page.
Logon > Logon Page	Provides a custom logon page that you can configure entirely from the properties screen.
Logon > Virtual Keyboard	Provides a configurable virtual keyboard for logon information entry.
Logon > VMware View Logon Page	Provides a custom logon page for VMware View.
Endpoint Security (Client-Side) > Anti-Spyware	Checks that the client is running specified anti-spyware software.
Endpoint Security (Client-Side) > Antivirus	Checks that the client is running specified antivirus software.
Endpoint Security (Client-Side) > Firewall	Checks that the client is running specified firewall software.
Endpoint Security (Client-Side) > Hard Disk Encryption	Checks that the client hard disk is encrypted.
Endpoint Security (Client-Side) > Linux File	Allows a check for a specific file with specified properties on a Linux system.
Endpoint Security (Client-Side) > Linux Process	Allows a check for a specific process on Linux systems.
Endpoint Security (Client-Side) > Mac File	Allows a check for a specific file with specified properties on Windows systems.
Endpoint Security (Client-Side) > Mac Process	Allows a check for a specific process on Windows systems.
Endpoint Security (Client-Side) > Machine Cert Auth	Allows a check for a machine certificate.
Endpoint Security (Client-Side) > Patch Management	Allows a check for patches to specific files.

Option	Description
Endpoint Security (Client-Side) > Peer-to-peer	Allows a check for peer to peer software on a system.
Endpoint Security (Client-Side) > Windows Cache and Session Control	Allows you to configure Windows clients to clean certain items after the session closes.
Endpoint Security (Client-Side) > Windows File	Allows a check for a specific file with specified properties on Windows systems.
Endpoint Security (Client-Side) > Windows Health Agent	Allows a check for a health agent on Windows systems.
Endpoint Security (Client-Side) > Windows Process	Allows a check for a specific process on Windows systems.
Endpoint Security (Client-Side) > Windows Protected Workspace	Allows configuration of a protected workspace in Windows.
Endpoint Security (Client-Side) > Windows Registry	Allows a check for a specific registry value in Windows.
General Purpose > Decision Box	Allows configuration of a choice of two branches for the user, with custom text describing each choice.
General Purpose > Email	Sends an email, when reached in the access policy.
General Purpose > iRule Event	Allows configuration of a choice of two branches for the user, with custom text describing each choice.
General Purpose > Local Database	Allows you to add entries to a local database.
General Purpose > Logging	Allows you to log a session variable result.
General Purpose > Message Box	Shows a message, and requires the user to click to continue.

A properties screen opens.

5. Configure the properties for the item.

6. Click **Save**.

The properties screen closes and the visual policy editor displays.

The access policy is configured with the empty action you have configured.

Click the **Apply Access Policy** link to apply and activate your changes to this access policy.

Adding an access policy assignment item

Before you start this task, configure an access profile.

Configure an access policy with an assignment action to assign a resource, local traffic pool, ACL, profile, or other item. Each assignment action works differently and assigns different items. Please read more about each item in the specific topic or online help.

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.

The visual policy editor opens the access policy in a separate screen.

- Click the (+) icon anywhere in the access policy to add a new action item.
A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
- Select an assignment action, then click **Add Item**:

Option	Description
Assignment > ACL Assign	Assigns an ACL to the access policy branch.
Assignment > Advanced Resource Assign	Directly assigns all types of resources.
Assignment > BWC Policy	Assigns a Bandwidth Controller policy to an access policy branch.
Assignment > Citrix Smart Access	Assigns a Citrix Smart Access filter to an access policy branch.
Assignment > Dynamic ACL	Assigns a dynamic ACL to an access policy branch.
Assignment > Resource Assign	Allows you to assign connection resources, remote desktops, and SAML resources.
Assignment > Route Domain and SNAT Selection	Allows you to assign a route domain, SNAT, and SNAT pool to an access policy branch.
Assignment > SSO Credential Mapping	Allows you to assign attributes for the SSO username and password.
Assignment > SWG Scheme Assign	Allows you to assign a specific Secure Web Gateway scheme.
Assignment > Webtop and Links Assign	Assigns a webtop and webtop links to an access policy branch.

A properties screen opens.

- Configure the properties for the item.
- Click **Save**.
The properties screen closes and the visual policy editor displays.

The access policy is configured with the assignment action you have configured.

Click the **Apply Access Policy** link to apply and activate your changes to this access policy.

Adding an access policy mapping item

Before you start this task, configure an access profile.

Configure an access policy with a mapping action to map resources or variables of one type to another type or value. Each mapping action works differently and assigns different items.

- On the Main tab, click **Access Policy > Access Profiles**.
The Access Profiles List screen opens.
- In the Access 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.
- Click the (+) icon anywhere in the access policy to add a new action item.
A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.
- Select a mapping action, then click **Add Item**:

Option	Description
Assignment > AD Group Resource Assign	Maps resources from an Active Directory group to access policy resources.
Assignment > LDAP Group Resource Assign	Maps resources from an LDAP group to access policy resources.
Assignment > Variable Assign	Allows you to assign predefined or custom variables to attributes, values, text, or expressions.

A properties screen opens.

5. For the Variable assign action, click the **Add new entry** button.
The AD and LDAP Group Assign actions already include an entry.
6. Click the **Edit** link.
7. Configure the settings for the assign action.
For the AD or LDAP group resource assign action, type the name of the group, then click **Add group manually**.
8. Configure the mapping items.
Refer to the specific documentation for each item to map items.
9. Click **Save**.
The properties screen closes and the visual policy editor displays.

The access policy is configured with the assignment action you have configured.

Click the **Apply Access Policy** link to apply and activate your changes to this access policy.

Chapter

3

Access Policy Item Reference

- *About logon items*
 - *About assignment items*
 - *About endpoint security client-side items*
 - *About endpoint security (server-side) access policy items*
 - *About general purpose items*
 - *About authentication items*
-

About logon items

Logon items either display on a logon screen, or specify and present a logon screen to a user.

Note: *Only the Virtual Keyboard item displays on a logon screen.*

Logon screens display input fields, and in some cases messages. The items that present a logon screen accept user input and store it in session variables for use in another access policy item; typically, that is an authentication item and it usually follows a logon item in an access policy.

When you work with a logon item, you can usually change some aspect of the logon experience.

Language on the screen

Access Policy Manager® (APM®) provides the text displayed on the logon screen translated into a number of languages. (Languages are specified in the access profile.) Selecting a language in a logon item translates the text to that language. Translated text can be used as is or customized further.

Text on the screen

APM provides default labels for the user input fields and for any messages that can be displayed on the logon screen. The text can be edited.

Fields on the screen

The logon page item provides up to five fields that can be displayed or not. The type for each field is user-selectable: password, text, select (from a list).

Some logon items include authentication-specific settings. These logon items are appropriate in specific cases only:

HTTP 401 error

The HTTP 401 Response logon item is appropriate in response to an HTTP 401 error. It can precede HTTP Basic or Kerberos authentication, or both.

HTTP 407 error

The HTTP 407 Response logon item is appropriate in response to an HTTP 407 error. It can precede HTTP Basic or Kerberos authentication, or both. It is applicable for use with Secure Web Gateway (SWG) explicit forward proxy only.

Standalone VMware View client

The VMware View logon page is for use with a standalone VMware View client. It presents a logon screen that is customized for the selected authentication type (from a set of supported types).

About the External Logon page

An External Logon page action provides a link to a logon page on an external server. An external solution can then provide robust logon credentials to the access policy. A logon action typically precedes the authentication action that checks the credentials provided on the logon page.

When an access policy reaches the External Logon page action:

- Access Policy Manager® sends an HTML page containing JavaScript code that redirects users to the external server.
- The client submits a `post_url` variable. This `post_url` variable is used by the external application to return a value to the access policy. When the user completes authentication on the external server, the external server posts back to the URL specified in this variable, to continue the session.

The value of `post_URL` is in the format: `http (or https) ://Access_Policy_Manager_URI/my.policy`. The `Access_Policy_Manager_URI` is the URI visible to the user, taken from the HTTP Host header value sent by the browser.

An External Logon Page action provides these configuration elements and options:

External Logon Server URI

Specifies the URI of the external logon server.

Split domain from full username

Specifies **Yes** or **No**.

- **Yes** - specifies that when a username and domain combination is submitted (for example, `marketing\jsmith` or `jsmith@marketing.example.com`), only the username portion (in this example, `jsmith`) is stored in the session variable `session.logon.last.username`.
- **No** - specifies that the entire username string is stored in the session variable.

About HTTP 401 Response

The HTTP 401 Response action sends an HTTP 401 Authorization Required Response page to capture HTTP Basic or Negotiate authentication. The HTTP 401 Response action provides three branches: Basic, Negotiate, and fallback. Typically, a basic type of authentication follows on the Basic branch and a Kerberos Auth action follows on the Negotiate branch. An HTTP 401 response action provides these configuration elements and options.

The action provides 401 response settings.

Basic Auth Realm

Specifies the authentication realm for use with Basic authentication.

HTTP Auth Level

Specifies the authentication required for the access policy.

- **none** - specifies no authentication.
- **basic** - specifies Basic authentication only.
- **negotiate** - specifies Kerberos authentication only.
- **basic+negotiate** - specifies either Basic or Kerberos authentication.

The action provides customization options that specify the text to display on the screen.

Language

Specifies the language to use to customize this HTTP 401 response page. Selecting a language causes the content in the remaining fields display in the selected language.

***Note:** Languages on the list reflect those that are configured in the access profile.*

Logon Page Input Field #1

Specifies the text to display on the logon page to prompt for input for the first field. When **Language** is set to **en**, this defaults to `Username`.

Logon Page Input Field #2

Specifies the text to display on the logon page to prompt for input for the second field. When **Language** is set to **en**, this defaults to `Password`.

HTTP response message

Specifies the text that appears when the user receives the 401 response, requesting authentication.

About HTTP 407 Response

The HTTP 407 response action sends an HTTP 407 Proxy Authentication Required page to capture HTTP Basic or Negotiate authentication. The HTTP 407 Response action provides three branches: Basic, Negotiate, and fallback. Typically, a basic type of authentication follows on the Basic branch and a Kerberos Auth action follows on the Negotiate branch. An HTTP 407 response action provides these configuration elements and options:

The action provides 407 response settings.

Basic Auth Realm

Specifies the authentication realm for use with Basic authentication.

HTTP Auth Level

Specifies the authentication required for the access policy.

- **none** - specifies no authentication.
- **basic** - specifies Basic authentication only.
- **negotiate** - specifies Kerberos authentication only.
- **basic+negotiate** - specifies either Basic or Kerberos authentication.

The action provides customization options that specify the text to display on the screen.

Language

Specifies the language to use to customize this HTTP 407 response page. Selecting a language causes the content in the remaining fields display in the selected language.

***Note:** Languages on the list reflect those that are configured in the access profile.*

Logon Page Input Field #1

Specifies the text to display on the logon page to prompt for input for the first field. When **Language** is set to **en**, this defaults to `Username`.

Logon Page Input Field #2

Specifies the text to display on the logon page to prompt for input for the second field. When **Language** is set to **en**, this defaults to `Password`.

HTTP response message

Specifies the text that appears when the user receives the 407 response, requesting authentication.

About logon page actions

A logon page action prompts for a user name and password or other identifying information. The logon page action typically precedes the authentication action that checks the credentials provided on the logon page. The logon page action provides up to five customizable fields and enables localization.

The logon page action provides these configuration options and elements.

Split domain from full username

Specifies **Yes** or **No**.

- **Yes** - specifies that when a username and domain combination is submitted (for example, `marketing\jsmith` or `jsmith@marketing.example.com`), only the username portion (in this example, `jsmith`) is stored in the session variable `session.logon.last.username`.
- **No** - specifies that the entire username string is stored in the session variable.

CAPTCHA configuration

Specifies a CAPTCHA configuration to present for added CAPTCHA security on the logon page.

Type

Specifies the type of logon page input field: **text**, **password**, **select**, **checkbox**, or **none**.

- **text** Displays a text field, and shows the text that is typed in that field.
- **password** Displays an input field, but displays the typed text input as asterisks.
- **select** Displays a list. The list is populated with values that are configured for this field.
- **checkbox** Displays a checkbox.
- **none** Specifies that the field is not displayed on the logon page.

Post Variable Name

Specifies the variable name that is prepended to the data typed in the text field. For example, the POST variable **username** sends the user name input `omaas` as the POST string `username=omaas`.

Session Variable Name

Specifies the session variable name that the server uses to store the data typed in the text field. For example, the session variable **username** stores the username input `omaas` as the session variable string `session.logon.last.username=omaas`.

Values

Specifies values for use on the list when the input field type is **select**.

Read Only

Specifies whether the logon page agent is read-only, and always used in the logon process as specified. You can use **Read Only** to add logon POST variables or session variables that you want to submit from the logon page for every session that uses this access policy, or to populate a field with a value from a session variable. For example, you can use the On-Demand Certificate agent to extract the **CN** (typically the user name) field from a certificate, then you can assign that variable to **session.logon.last.username**. In the logon page action, you can specify `session.logon.last.username` as the session variable for a read only logon page field that you configure. When Access Policy Manager® displays the logon page, this field is populated with the information from the certificate **CN** field (typically the user name).

Additionally, customization options specify text and an image to display on the screen.

Language

Specifies the language to use to customize this logon page. Selecting a language causes the content in the remaining fields display in the selected language.

***Note:** Languages on the list reflect those that are configured in the access profile.*

Form Header Text

Specifies the text that appears at the top of the logon box.

Logon Page Input Field # *number*

Specifies the text to display for each input field (number 1 through 5) that is defined in the Logon Page Agent area with **Type** set to other than **none**.

Logon Button

Specifies the text that appears on the logon button, which a user clicks to post the defined logon agents.

Front Image

Specifies an image file to display on the logon page. The **Replace Image** link enables customization and the **Revert to Default Image** discards any customization and use the default logon page image.

Save Password Check Box

Specifies the text that appears adjacent to the check box that allows users to save their passwords in the logon form. This field is used only in the secure access client, and not in the web client.

New Password Prompt

Specifies the prompt displayed when a new Active Directory password is requested.

Verify Password Prompt

Specifies the prompt displayed to confirm the new password when a new Active Directory password is requested.

Password and Password Verification do not Match

Specifies the prompt displayed when a new Active Directory password and verification password do not match.

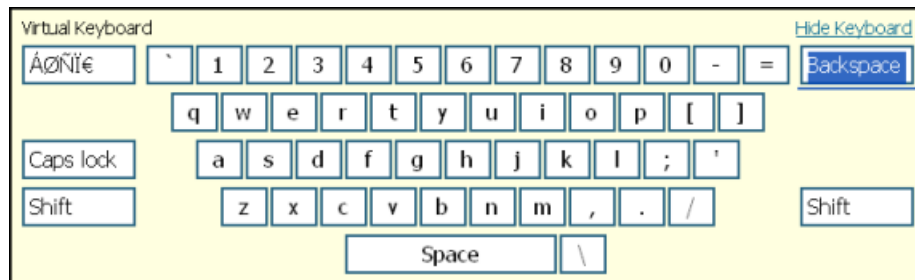
Don't Change Password

Specifies the prompt displayed when a user should not change password.

About the virtual keyboard

A virtual keyboard displayed on the logon screen prevents password characters from being typed on the physical keyboard. The virtual keyboard appears on the logon screen when a user clicks in the password field. A user then types the password by clicking the characters on the virtual keyboard, instead of typing them on the physical keyboard.

A virtual keyboard action applies to all logon page actions that follow it in the access policy.



The virtual keyboard action provides these configuration elements and options:

Virtual Keyboard

Specifies whether the onscreen virtual keyboard is enabled or disabled.

Move Keyboard After Every Keystroke

Specifies whether the onscreen keyboard moves after the user enters a keystroke with a mouse click.

Allow Manual Input

Specifies whether a user can type the password with the physical keyboard, in addition to clicking keys on the virtual keyboard.

About the VMware View logon page action

A VMware View logon page action can display a message or can request Windows, RSA SecurID, or RADIUS logon credentials. A logon action typically precedes the authentication action that checks the credentials provided on the logon page.

The VMware View logon page provides these configuration elements and options:

VMware View logon screen

Specifies the type of logon screen to display:

- **Windows Password** - requests Windows logon credentials.
- **RSA SecurID** - requests RSA SecurID logon credentials.
- **RADIUS** - requests RADIUS credentials.
- **Disclaimer** - displays a message dialog box; for example, to display acceptable use terms.

VMware View Windows Domains

Specifies domain names separated by commas; for use with the Windows Password screen.

VMware View RADIUS Auth Label

Specifies the name of the RADIUS authentication provider to display on the RADIUS logon screen in a message similar to this one: Please provide your *Auth Label* credentials

The VMware View logon page action also provides customization options to specify the text to display on the screen:

Language

Specifies the language to use to customize this logon page. Selecting a language causes the content in the remaining fields display in the selected language.

***Note:** Languages on the list reflect those that are configured in the access profile.*

Logon Page Input Field #1

Specifies the text to display on the logon page to prompt for input for the first field. When **Language** is set to **en**, this defaults to `Username`.

Logon Page Input Field #2

Specifies the text to display on the logon page to prompt for input for the second field. When **Language** is set to **en**, this defaults to `Password`.

Disclaimer message

Specifies a message to display in the disclaimer logon screen.

About assignment items

Most assignment items support assigning resources to a session. In contrast, the Variable Assign item supports assigning values to existing variables, to existing configuration elements, and to variables that you define yourself.

Resource assignment

A resource assignment item is usually placed immediately prior to an **Allow** ending on a branch in an access policy. At that point, any branching (based on client type or geolocation, and so on), client software checks, SSL client certificate checks, and authentication items are complete. Resource assignment supports:

- Selection of the resources that are needed to establish a network access, portal access, or application access session, including a webtop and any ACLs.
- Mapping resources to an Active Directory or LDAP group.
- Overriding the pool assignment made in a virtual server.

Resource assignment also supports selection of resources for bandwidth control, enforcement of Secure Web Gateway (SWG) schemes, and so on.

Variable assignment

A variable assignment item is placed where needed in an access policy. Variable assignment items can support:

- Replacing the value of one configuration object, such as a subnet, with another configuration object of the same type.
- Replacing the value of a session variable.
- Taking a value from an AAA attribute (which must be already available in the session, retrieved by another item), and assigning the attribute value to a variable.
- Creating a variable for any reason (for example, storing a value for later retrieval or using the value in arithmetic operations).
- Using Tcl expressions to derive values and assign them to variables.

About ACL Assign

An ACL Assign action dynamically assigns static access control lists (ACLs). ACLs then apply only to clients that reach such an assignment action in the access policy. An ACL Assign action provides these configuration elements and options: selection of static ACLs from those configured in Access Policy Manager®.

When no ACLs are assigned in an access policy, the default behavior allows access. When an ACL is assigned in an access policy, it can restrict resources to only those specified in the ACL provided that the last ACE in the list is configured to reject any connection not matched by a previous entry.

Note: The Advanced Resource Assign action also supports ACL assignment.

About AD Group Resource Assign

The AD Group Resource Assign action enables users to create entries that specify Active Directory groups and assign resources to them.

An AD Group Resource Assign action provides these configuration elements and options:

Groups

Specifies the AD groups to which resources are assigned. A list of groups can be imported through the AD AAA server and created manually by typing group names.

Resources

Specifies Static ACLS, Network Access resources, App Tunnels, and so on to assign to the selected groups. Any resource on the system can be assigned to a group. The system limits apply; for example, only one webtop should be assigned to a group.

About Advanced Resource Assign

The Advanced Resource Assign action enables assignment of resources.

An Advanced Resource Assign action provides these configuration elements and options:

Resource type

Specifies a type of resource, one-per-tab, and, on each tab, provides a check list or radio button list of such resources for selection. Resource types include: Network Access, Portal Access, App Tunnels, Remote Desktops, Static ACLs, SAML, Webtops, Webtop Links, and Static Pools.

About BWC Policy

The BWC Policy action enables users to assign bandwidth control (BWC) policies to the traffic that passes through the virtual server.

A BWC Policy action provides these configuration elements and options:

Static BWC policies

Specifies one or more static BWC policies from those configured on the BIG-IP® system.

Dynamic BWC Policy

Specifies the name of one dynamic BWC policy that was configured to shape traffic for Citrix clients that support MultiStream ICA.

Very High Citrix BWC Category

Specifies the name of the category in the BWC policy that assigns a percentage of the maximum bandwidth to a very high level of Citrix traffic.

High Citrix BWC Category

Specifies the name of the category in the BWC policy that assigns a percentage of the maximum bandwidth to a high level of Citrix traffic.

Very High Citrix BWC Category

Specifies the name of the category in the BWC policy that assigns a percentage of the maximum bandwidth to a medium level of Citrix traffic.

Very High Citrix BWC Category

Specifies the name of the category in the BWC policy that assigns a percentage of the maximum bandwidth to a low level of Citrix traffic.

Note: For more information, refer to *BIG-IP® Access Policy Manager®: Third-Party Integration Implementations on the AskF5™ web site* (<http://support.f5.com/kb/en-us.html>).

About Citrix Smart Access

The Citrix Smart Access action enables users to assign Citrix SmartAccess filters to the session. A filter is a name; it is defined in a Citrix software product.

A Citrix Smart Access action provides these configuration elements and options:

Assignment

One or more entries each of which specifies the name of a filter. The name must match the name that is specified in the Citrix software product.

For more information, refer to *BIG-IP® Access Policy Manager®: Third-Party Integration Implementations on the AskF5™ web site* (<http://support.f5.com/kb/en-us.html>).

About Dynamic ACL

A *dynamic ACL* is an ACL that is created on and stored in an LDAP, RADIUS, or Active Directory server. A Dynamic ACL action dynamically creates ACLs based on attributes from the AAA server. Because a dynamic ACL is associated with a user directory, this action can assign ACLs specifically per the user session.

Note: Access Policy Manager® supports dynamic ACLs in an F5® ACL format, and in a subset of the Cisco ACL format.

A Dynamic ACL action provides these configuration elements and options:

Source

Specifies an option and the attribute from which the Dynamic ACL action extracts ACLs: **Custom** indicates an F5 ACL from an Active Directory, RADIUS, or LDAP directory; **Cisco AV-Pair VSA** indicates a Cisco AV-Pair ACL from a RADIUS directory; the field is prepopulated with:
`session.radius.last.attr.vendor-specific.1.9.1.`

ACL

Specifies the dynamic ACL container configured on the BIG-IP® system.

Format

Specifies the format (F5 or Cisco) in which the ACL is specified.

Note: To succeed, a Dynamic ACL action must follow an authentication or query action to capture the authentication variables that contain the dynamic ACL specification.

About LDAP Group Resource Assign

The LDAP Group Resource Assign action enables users to create entries that specify LDAP groups and assign resources to them.

An LDAP Group Resource Assign action provides these configuration elements and options:

Groups

Specifies the LDAP groups to which resources are assigned. A list of groups can be imported through the LDAP AAA server and created manually by typing group names.

Resources

Specifies Static ACLS, Network Access resources, App Tunnels, and so on to assign to the selected groups. Any resource on the system can be assigned to a group. The system limits apply; for example, only one webtop should be assigned to a group.

About Pool Assign

The Pool Assign action can dynamically assign a local traffic pool, enabling pool selection based on the result of prior access policy action results. An Pool Assign action provides this configuration element only: selection of a static pool. However, this assignment occurs only when another pool assignment does not take higher priority.

Pool assignment priority

A pool is selected from among valid pools in this priority order:

- A pool selected by an iRule that is defined for the virtual server takes precedence over any other.
- A static pool defined in the Pool Assign action takes precedence over a static pool defined for the virtual server.
- A static pool defined for the virtual server takes lowest precedence.

About resource assignment

You can assign access control lists, network access resources, portal access resources, app tunnel resources, a webtop, and webtop links to the access policy using one of the resource assign actions. Each resource assign action provides a similar function, with the following differences.

Advanced resource assign

Allows you to assign all resources: network access, portal access, app tunnels, remote desktops, ACLs, SAML resources, webtops, webtop links, and local traffic pools.

Resource assign

Assigns connection resources only: network access, portal access, app tunnels, remote desktops, and SAML resources.

ACL assign

Assigns static ACLs only.

Webtop and links assign

Assigns a webtop and webtop links only.

Any resource you want to assign to an access policy branch must be added with either an advanced resource assign action, or with the specific resource assign action for that type of resource. You must assign a network access resource for a network access connection. For portal access, app tunnels, or remote desktops, you must assign the appropriate tunnels. You can assign a webtop specific to network access or portal access, or you can assign a full webtop that can display multiple resources and links. You assign ACLs to all access types with the advanced resource assign action or with the ACL assign action.

Note: For a web access management connection, you do not assign a connection resource or a webtop.

About Route Domain and SNAT Selection

The Route Domain and SNAT Selection action enables dynamic assignment of a route domain and of SNAT.

A Route Domain and SNAT Selection action provides these configuration elements and options:

Route Domain

Specifies a route domain. Enables route domain-based policy routing, sending a user to another route domain based on the outcomes of previous branches in the access policy.

SNAT

Specifies a SNAT to provide secure network address translation (SNAT) to the self IP address of the BIG-IP® device, or to choose from a pool of configured internal addresses for SNAT. SNAT precedence is determined according to the following rules:

- First, if a SNAT is defined in a Network Access resource configuration, APM uses that SNAT.
- If there is no SNAT defined in the Network Access resource, or the resource is another type, the APM takes the SNAT from this assignment in the access policy.
- If there is no SNAT assigned in the access policy, the APM uses the SNAT from the virtual server definition.

About SSO Credential Mapping

The SSO Credential Mapping action caches the user name and password for use with single sign-on (SSO) applications in the enterprise. This action enables users to forward stored user names and passwords to applications and servers automatically, without having to input credentials repeatedly.

The SSO Credential Mapping action provides these configuration elements and options.

SSO Token Username

One of these:

- **Username from Logon Page** - when selected, the Tcl expression that APM[®] uses to obtain the username from session variables displays; it is read-only.
- **sAMAccountName from Active Directory** - when selected, the Tcl expression that APM uses displays; it is read-only.
- **sAMAccountName from LDAP Directory** - when selected, the Tcl expression that APM uses displays; it is read-only.
- **Custom** - when selected, the last-displayed Tcl expression remains in the entry field. This field can be edited; another Tcl expression can be entered.

SSO Token Password

One of these:

- **Password from Logon Page** - when selected, the Tcl expression that APM uses to obtain the username from session variables displays; it is read-only.
- **Custom** - when selected, the last-displayed Tcl expression remains in the entry field. This field can be edited.

About SWG Scheme Assign

The SWG Scheme Assign action can dynamically assign a Secure Web Gateway (SWG) scheme to a session. An SWG scheme specifies URL filters to apply to an SWG forward proxy configuration.

An SWG Scheme Assign action provides these configuration elements and options:

SWG Scheme

Specifies the name of one SWG scheme that is configured on the BIG-IP[®] system.

About Variable Assign

The Variable Assign action can includes one or more entries. An entry specifies a variable and assigns a value to it.

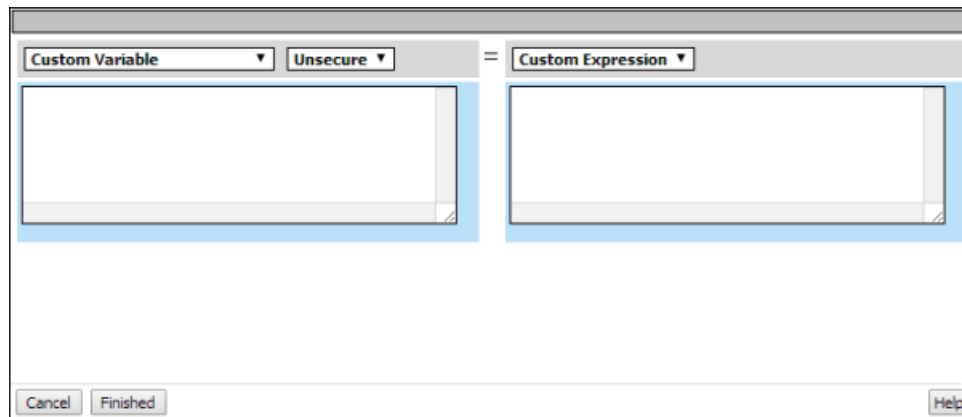


Figure 3: A variable assign entry screen as it displays initially

In the entry screen, the variable is specified in the left pane and the value is specified in the right pane.

A Variable Assign action provides these configuration elements and options for the variable:

Custom Variable

Specifies a variable name. It can be any name including the name of session variable.

Predefined Session Variable

Specifies a session variable name which must be selected from the **Variable** list.

Unsecure or Secure

Specifies whether the variable is secure. A secure variable is stored in encrypted form in the session database. The value of a secure variable is not displayed in the session report, or logged by the logging agent.

An Variable Assign action provides these configuration elements and options for the value:

Custom Expression

Specifies a Tcl expression. The result of the expression is used as the value.

AAA attribute

Specifies the name of the attribute that contains the value:

- **Agent Type** - specifies the type of AAA server: AD, LDAP, or RADIUS.
- **Attribute Type** - specifies the attribute type to use: LDAP, RADIUS or, for the AD agent type, one or these:
 - **Use user's attribute**
 - **Use user's primary group attribute**
- **Agent type attribute name** - specifies the name of the attribute that contains the value.

Text

Specifies a text string to use as the value. The text entered in this field is used as is.

Session Variable

Specifies the name of a session variable from which to get the value.

About Webtop and Links

The Webtop and Links action can assign a webtop or preconfigured webtop links or both to a session.

A Webtop and Links action provides these configuration elements and options:

Webtop Links

Specifies one or more webtop links.

***Note:** Webtop links apply only to a full webtop.*

Webtop

Specifies one webtop. This can be a full webtop, portal access webtop, or a network access webtop.

About endpoint security client-side items

Endpoint security is a strategy for ensuring that a client device does not present a security risk before it is granted a remote access connection to the network.

Endpoint security verifies that desktop antivirus and firewall software is in place, systems are patched, keyloggers or other dangerous processes are not running, and sensitive data is not left behind in web caches and other vulnerable locations.

Configuring endpoint security (client-side) access policy items enables verification actions and other security-enhancing actions:

- On a Linux, Mac, or Windows client, client-side items can confirm that software meets requirements and can confirm the presence or absence of files and processes.
- On a Windows client, client-side items can confirm the registry, open a protected workspace, or perform cache and session control.

About client-side action requirements and alternatives

Endpoint security (client-side) access policy items require installation of client components. Access Policy Manager® uses ActiveX controls or browser plug-ins to collect information about client systems.

Not all clients support browser add-ons or allow browser software installation. For these clients, the server-side security process can inspect HTTP headers to gather information about the client operating system and browser type. The server-side Client Capability action determines whether a client is capable of running client-side actions.

About the Anti-spyware action

The Anti-spyware action checks for anti-spyware software on a client computer. When checking for multiple anti-spyware types, if one anti-spyware type matches the software on the client system, the action passes, regardless of other anti-spyware conditions that are specified in the item.

An Anti-spyware action provides these settings and options:

Continuously check the result and end the session if it changes

Specifies **Enabled** or **Disabled**.

Platform

Specifies a platform. The default is **Any**. When a platform is selected, the Vendor ID and Product ID lists update to include the products and vendors that are supported for that platform according to the EPSEC package that is installed on the BIG-IP® system.

***Note:** A link to a report that includes the anti-spyware software that Access Policy Manager® currently supports is available on the BIG-IP system Welcome page.*

Vendor ID

Specifies a vendor ID (from the list of supported vendors) or **Any**.

Product ID

Specifies a product ID (from the list of supported products) or **Any**.

State

Specifies one of these states:

- **Enabled** When selected, the action verifies that the anti-spyware software is enabled
- **Disabled** When selected, the action verifies that the anti-spyware software is disabled.
- **Unspecified** When selected, the action does not verify the state of the software.

Engine Version

Specifies the engine version number; when specified the Anti-spyware action verifies this information.

DB Version

Specifies the database version number; when specified the Anti-spyware action verifies this information.

DB Age Not Older Than (days)

Specifies the database age in days; when specified the Anti-spyware action verifies this information.

Last Scan Time Not Older Than (days)

Specifies a number of days; when specified the Anti-spyware action verifies that the last scan did not occur more than the specified number of days ago.

About the Antivirus action

The Antivirus action checks for antivirus software on the client computer. When checking for multiple antivirus types, if one antivirus type matches the software on the client system, the action passes, regardless of other antivirus conditions that are specified in the action.

An Antivirus action provides these settings and options:

Continuously check the result and end the session if it changes

Specifies **Enabled** or **Disabled**.

Platform

Specifies a platform. The default is **Any**. When a platform is selected, the Vendor ID and Product ID lists update to include the products and vendors that are supported for that platform according to the EPSEC package that is installed on the BIG-IP® system.

***Note:** A link to a report that includes the antivirus software that Access Policy Manager® currently supports is available on the BIG-IP system Welcome page.*

Vendor ID

Specifies a vendor ID (from the list of supported vendors) or **Any**.

Product ID

Specifies a product ID (from the list of supported products) or **Any**.

State

Specifies one of these states:

- **Enabled** - when selected, the action verifies that the antivirus software is enabled
- **Disabled** - when selected, the action verifies that the antivirus software is disabled.
- **Unspecified** - when selected, the action does not verify the state of the software.

Version

Specifies a version; when specified, the antivirus action verifies the version of the software.

Engine Version

Specifies the engine version number; when specified, the antivirus action verifies this information.

DB Version

Specifies the database version number; when specified, the antivirus action verifies this information.

DB Age Not Older Than (days)

Specifies the database age in days; when specified, the antivirus action verifies this information.

Last Scan Time Not Older Than (days)

Specifies a number of days; when specified, the antivirus action verifies that the last scan did not occur more than the specified number of days ago.

About the Firewall action

The Firewall action checks for firewall software on the client computer. When this action includes checks for multiple firewall types, if one firewall type matches the software on the client computer, the action passes, regardless of other firewall conditions that are specified in the action.

A firewall action provides these settings and options:

Continuously check the result and end the session if it changes

Specifies **Enabled** or **Disabled**.

Platform

Specifies a platform. The default is **Any**. When a platform is selected, the Vendor ID and Product ID lists update to include the products and vendors that are supported for that platform according to the EPSEC package that is installed on the BIG-IP® system.

***Note:** A link to a report that includes the firewall software that Access Policy Manager® currently supports is available on the BIG-IP system Welcome page.*

Vendor ID

Specifies a vendor ID (from the list of supported vendors) or **Any**.

Product ID

Specifies a product ID (from the list of supported products) or **Any**.

State

Specifies one of these states:

- **Enabled** When selected, the action verifies that the firewall software is enabled
- **Disabled** When selected, the action verifies that the firewall software is disabled.
- **Unspecified** When selected, the action does not verify the state of the software.

Version

Specifies a version; when specified, the firewall action verifies the version of the software.

About Hard Disk Encryption

The Hard Disk Encryption action checks for hard disk encryption software on a client computer. When this action includes checks for multiple hard disk encryption types, if one of the specified hard disk encryption types matches the software on the client system, the action passes, regardless of other hard disk encryption conditions that are specified in the item.

A Hard Disk Encryption action provides these settings and options:

Continuously check the result and end the session if it changes

Specifies **Enabled** or **Disabled**.

Platform

Specifies a platform. The default is **Any**. When a platform is selected, the Vendor ID and Product ID lists update to include the products and vendors that are supported for that platform according to the EPSEC package that is installed on the BIG-IP® system.

***Note:** A link to a report that includes the hard disk encryption software that Access Policy Manager® currently supports is available on the BIG-IP system Welcome page.*

Vendor ID

Specifies a vendor ID (from the list of supported vendors) or **Any**.

Product ID

Specifies a product ID (from the list of supported products) or **Any**.

Encryption State

Specifies one of these states:

- **Enabled** When selected, the action verifies that all disk volumes are encrypted on the client.
- **Disabled** When selected, the action verifies all disk volumes are not encrypted on the client.
- **Unspecified** When selected, the action verifies that hard disk encryption software is installed on the client.

Version

Specifies a version; when specified, the Hard Disk Encryption action verifies the version of the software.

About Linux File

The Linux File action can verify the presence of specific files and can verify one or more file properties in situations where doing so increases confidence in the security of the client system. If a file with the described properties exists, the access policy passes the client to the successful branch. If the file does not exist, or a file exists but one or more properties are not correct, the access policy passes the client to the fallback branch.

The Linux File action provides the following configuration elements and options:

Continuously check the result and end the session if it changes

Specifies **Enabled** or **Disabled**.

FileName

Specifies the file name for which to check; for example `csound`.

MD5

Specifies the MD5 checksum. An MD5 checksum provides easily computable verification of the identity of a file using a cryptographic hash algorithm. The MD5 checksum is a 32-digit hexadecimal value. For example, the checksum for a zero-byte file is always d41d8cd98f00b204e9800998ecf8427e.

Size

Specifies the size of the file in bytes. The default value is 0 which is the same as not specifying a size; a size of zero (0) is not verified.

Note: A zero-byte file is specified with the MD5 checksum for a zero-byte file in the **MD5** field.

Date

Specifies the file last modified date.

Note: The date must be translated first to GMT, and then to a 24-hour clock.

About Linux Process

The Linux Process action can verify that one or more particular processes are or are not running on a client system.

The Linux Process action provides these configuration elements and options:

Continuously check the result and end the session if it changes

Specifies **Enabled** or **Disabled**.

Expression

Specifies a Boolean expression to use to check for a process. The expression can include these wildcards: * and ?, and parentheses () to combine values, and the logical operators AND, OR, and NOT. This is the syntax for a process check expression: "process name" | (EXPRESSION) | NOT EXPRESSION | EXPRESSION AND EXPRESSION | EXPRESSION OR EXPRESSION

Note: Double quotes (" ") are required around each process name.

Here is an example expression: "httpd" AND NOT "smtpd". Using this expression, the Linux Process action verifies that the HTTP daemon (httpd) is running on the system, and that the SMTP daemon (smtpd) is not running. Using another example expression, ("process1" OR "process2") AND "process3*", the action verifies the presence of either process1 or process2, and a process with a name that is process3 or starts with process3.

About Mac File

The Mac File action can verify the presence of specific files and can verify one or more file properties in situations where doing so increases confidence in the security of the client system. If a file with the described properties exists, the access policy passes the client to the successful branch. If the file does not exist, or a file exists but one or more properties are not correct, the access policy passes the client to the fallback branch.

The Mac File action provides the following configuration elements and options:

Continuously check the result and end the session if it changes

Specifies **Enabled** or **Disabled**.

FileName

Specifies the file name for which to check; for example `check.txt`.

MD5

Specifies the MD5 checksum. An MD5 checksum provides easily computable verification of the identity of a file using a cryptographic hash algorithm. The MD5 checksum is a 32-digit hexadecimal value. For example, the checksum for a zero-byte file is always `d41d8cd98f00b204e9800998ecf8427e`.

Size

Specifies the size of the file in bytes. The default value is 0 which is the same as not specifying a size; a size of zero (0) is not verified.

***Note:** A zero-byte file is specified with the MD5 checksum for a zero-byte file in the **MD5** field.*

Date

Specifies the file last modified date.

***Note:** The date must be translated first to GMT, and then to a 24-hour clock.*

About Mac Process

The Mac Process action can verify that one or more particular processes are or are not running on a client system.

The Mac Process action provides these configuration elements and options:

Continuously check the result and end the session if it changes

Specifies **Enabled** or **Disabled**.

Expression

Specifies a Boolean expression to use to check for a process. The expression can include these wildcards: * and ?, and parentheses () to combine values, and the logical operators AND, OR, and NOT. This is the syntax for a process check expression: "process name" | (EXPRESSION) | NOT EXPRESSION | EXPRESSION AND EXPRESSION | EXPRESSION OR EXPRESSION

***Note:** Double quotes (" ") are required around each process name.*

Here is an example expression: "httpd" AND NOT "smtpd". Using this expression, the Mac Process action verifies that the HTTP daemon (httpd) is running on the system, and that the SMTP daemon (smtpd) is not running. Using another example expression, ("process1" OR "process2") AND "process3*", the action verifies the presence of either process1 or process2, and a process with a name that is process3 or starts with process3.

About Machine Cert Auth

A Machine Certificate Auth action can check for the existence of fields in a machine certificate to ensure that Windows and Mac client systems comply with your security policy.

Table 1: Client-specific requirements

Client	Description
Windows	The Machine Cert Auth action accesses the machine certificate private key; admin privilege is required to do this. A user that runs without admin privilege cannot successfully run this check unless the machine certificate checker service is installed on the machine. (The Machine Certificate Checker Service is available for inclusion in the Windows client package from the Secure Connectivity area of Access Policy Manager.)
Mac	The Machine Cert Auth action accesses the machine certificate private key. If the certificate is stored in a keychain other than user's own keychain, such as the system keychain, then an ACL is required for non-admin users to be able to access this private key.

The Machine Certificate Auth action provides the following configuration elements and options:

Certificate Store Name

Specifies the certificate store name that the action attempts to match. The certificate store can be a system store with a predefined name, such as MY, or a user-defined name. The store name can contain alphanumeric characters. The Machine Cert Auth action treats MY as the default store name for both Mac and Windows clients.

Certificate Store Location

Specifies the type and location of the store that contains the certificate, either the local machine or the current user. For a Windows client, the store locations are in the following registry locations:

- **LocalMachine** When specified, the action searches in `HKEY_LOCAL_MACHINE` for the machine certificate.
- **CurrentUser** When specified, the action searches in `HKEY_CURRENT_USER` for the machine certificate.

For a Mac client, the store locations are keychains in the following domains:

- **LocalMachine** When specified, the action searches in the keychain specified in **Certificate Store Name** in the system preference domain.
- **CurrentUser** When specified, the action searches in the keychain specified in **Certificate Store Name** in the user preference domain.

For a Mac client, the following examples apply.

- If **Certificate Store Name** is set to `System.keychain` and **Certificate Store Location** is set to **LocalMachine**, the action searches for the machine certificate in `/Library/Keychains/System.keychain`.
- If **Certificate Store Name** is set to `login.keychain` and **Certificate Store Location** is set to **CurrentUser**, the action searches for the machine certificate in `/Library/Keychains/login.keychain` and then searches for the machine certificate in `/Users/username/Library/Keychains/login.keychain`.
- If **Certificate Store Name** is set to MY then the action searches for the machine certificate in the default keychain of **Certificate Store Location**.

CA Profile

Specifies the certificate authority profile for the particular machine certificate.

OCSP Responder

Specifies an AAA OCSP responder (configured in Access Policy Manager®) to provide certificate status. The OCSP responder checks the status of the machine certificate configured in the Machine Cert Auth action.

Save Certificate in a session variable

Specifies **Enabled** or **Disabled**. When **Enabled**, specifies that the complete encrypted text of the machine certificate be saved in a session variable, `session.check_machinecert.<name>.cert.cert`.

Allow User Account Control right elevation prompts

Specifies **Yes** or **No**. When set to **Yes**, suppresses the UAC prompt during private key checking for non-admin users.

Match Subject CN with FQDN

Specifies **Yes** or **No**. When set to **Yes**, specifies that the common name in the machine certificate matches the computer's fully qualified domain name (FQDN) such as,
`CHR-L-SMITH2.MARKETING.SITEREQUEST.COM`.

Match subject Alt Name with FQDN

Specifies a regular expression used to extract content from the first subgroup matched in the Subject Alternative Name, and then to compare the extracted content with the machine's FQDN.

***Note:** The order of RDNs is the same as is displayed; the required separator is a comma , .*

Here are some examples of regex extraction.

- Partial extraction. For example, `.*DNS Name=([^\,]+).*` or `.*Other Name:Principal Name=([^\,]+).*`. For a regular expression `.*DNS Name=([^\,]+).*`, the value of the DNS Name field is extracted for matching.
- Whole extraction. Using `(.*)` specifies that the entire SubjectAltName content be extracted for matching.

Match Issuer

Specifies a regular expression that is used to match the Issuer content against the specified pattern.

***Note:** The order of RDNs is the same as is displayed; the required separator is a comma , .*

Here are some examples of regex extraction.

- Partial match. `CN=.*, OU=FP, O=F5, L=San Jose, S=CA, C=US`
- Exact match. `E=test@f5.com, CN=f5clientrootcert, OU=es, O=f5, L=london, S=chertsey, C=uk`

Match Serial Number

Specifies a serial number that must be an exact match for the certificate serial. The hex string must be specified in the same order as it is displayed by OpenSSL and Windows certificate tools. For example, `33:AA:7B:82:00:01:00:00:00:33`.

About Machine Info

The Machine Info action retrieves MAC addresses for network adapters on Mac, Linux, and Windows clients. It retrieves additional information on Windows clients

After retrieving the information, the Machine Info action creates session variables and stores the values in them. Session variables can be used in Tcl expressions and are also available for configuring an expression using the expression builder pull-down menu item **Machine Info**.

***Note:** In a session variable value, any special characters are represented by ASCII characters. For example, a space character is represented by the value `%20`. Leading and trailing white space characters are removed.*

The Machine Info action collects the following information and creates the following session variables.

Information	Session variable name
CPU Name	session.machine_info.cpu.name
CPU Vendor ID	session.machine_info.cpu.vendor
CPU Description	session.machine_info.cpu.description
CPU maximum clock	session.machine_info.cpu.max_clock
Motherboard manufacturer	session.machine_info.motherboard.manufacturer
Motherboard serial number	session.machine_info.motherboard.sn
Motherboard product	session.machine_info.motherboard.product
BIOS manufacturer	session.machine_info.bios.manufacturer
BIOS serial number	session.machine_info.bios.sn
BIOS version	session.machine_info.bios.version
Number of network adapters	session.machine_info.net_adapter.count
First network adapter name	session.machine_info.net_adapter.list.0.name
Second network adapter name	session.machine_info.net_adapter.list.1.name
First network adapter MAC address (Collected from Linux, Mac, and Windows clients)	session.machine_info.net_adapter.list.0.mac_address
Second network adapter MAC address (Collected from Linux, Mac, and Windows clients)	session.machine_info.net_adapter.list.1.mac_address
Number of hard drives	session.machine_info.hdd.count
First hard drive model number	session.machine_info.hdd.list.0.model
Second hard drive model number	session.machine_info.hdd.list.1.model
First hard drive serial number	session.machine_info.hdd.list.0.sn
Second hard drive serial number	session.machine_info.hdd.list.1.sn

About Patch Management

The Patch Management action can check for patch management software on the client system. When this action includes checks for multiple patch management types, if one specified type matches, the action passes, regardless of other conditions that are specified in the action.

The Patch Management action provides the following configuration elements and options:

Continuously check the result and end the session if it changes

Specifies **Enabled** or **Disabled**.

Platform

Specifies a platform. The default is **Any**. When a platform is selected, the Vendor ID and Product ID lists update to include the products and vendors that are supported for that platform according to the EPSEC package that is installed on the BIG-IP® system.

Note: A link to a report that includes the antivirus software that Access Policy Manager® currently supports is available on the BIG-IP system Welcome page.

Vendor ID

Specifies a vendor ID (from the list of supported vendors) or **Any**.

Product ID

Specifies a product ID (from the list of supported products) or **Any**.

Automatic Updates

Specifies one of these values:

- **Enabled** When selected, the action verifies that patch management software is running on the client system.
- **Disabled** When selected, the action verifies that patch management software is not running on the client system.
- **Unspecified** When selected, the action does not perform either verification.

Version

Specifies a version; when specified, the Patch Management action verifies the version of the software.

Max Allowed No. of Missing Critical Updates

Specifies a number; when specified, the action verifies that the number of missing critical updates for the software is less than this number.

About Peer-to-Peer

The Peer-to-Peer action checks for peer-to-peer software on the client system.

The Peer-to-Peer action provides these configuration elements and options:

Continuously check the result and end the session if it changes

Specifies **Enabled** or **Disabled**.

Check for software in the list, and

Specifies one of these options:

- **pass if at least one listed software matches** When selected, the action sends traffic to the successful branch if at least one software item in the list matches the software that is present on the client system.
- **fail if unlisted software found** -When selected, the action sends traffic to the fallback branch when any software that is not included in the list is found on the system. In this case, the list functions as a whitelist; if any endpoint software is found on the client system that is not included in the list, the check fails and traffic goes to the fallback branch.
- **fail if any listed software matches** When selected, the action sends traffic to the fallback branch when any software item in the list is found on the client system. In this case, the list functions as a blacklist.

Platform

Specifies a platform. The default is **Any**. When a platform is selected, the Vendor ID and Product ID lists update to include the products and vendors that are supported for that platform according to the EPSEC package that is installed on the BIG-IP® system.

***Note:** A link to a report that includes the peer-to-peer software that Access Policy Manager® currently supports is available on the BIG-IP system Welcome page.*

Vendor ID

Specifies a vendor ID (from the list of supported vendors) or **Any**.

Product ID

Specifies a product ID (from the list of supported products) or **Any**.

State

Specifies one of these values:

- **Enabled** When selected, the action verifies that peer-to-peer software is running on the client system.
- **Disabled** When selected, the action verifies that peer-to-peer software is not running on the client system.
- **Unspecified** When selected, the action does not verify the state.

Version

Specifies a version; when specified, the Peer-to-Peer action verifies the version of the software.

About Windows Cache and Session Control

The Windows Cache and Session Control action can clean up after and control a session in a number of ways.

***Note:** The Windows Cache and Session Control action, and the Windows Protected Workspace action, are not compatible and should not be used in the same session.*

The Windows Cache and Session Control action provides these configuration elements and options:

Clean temporary Internet files and cookies

Specifies **Disabled** or **Enabled**. When set to **Enabled**, the action deletes temporary files and cookies after logout.

Clean forms and passwords autocomplete data

Specifies **Disabled** or **Enabled**. When set to **Enabled**, the action clears autocomplete entries in forms and fields after logout.

Empty Recycle Bin

Specifies **Disabled** or **Enabled**. When set to **Enabled**, the action empties the system Recycle Bin after logout.

Force session termination if the browser or Webtop is closed

Specifies **Disabled** or **Enabled**. When set to **Enabled**, the action forces the session to terminate after the browser or Webtop is closed.

Remove dial-up entries used by Network Access client

Specifies **Disabled** or **Enabled**. When set to **Enabled**, the action removes dial-up networking entries after logout.

Terminate session on User Inactivity

Specifies **Disabled** or *n minutes*, or *n hours* or **Custom** and a number of minutes. When not set to **Disabled**, the action terminates the session after the specified amount of time elapses.

Lock workstation on User Inactivity

Specifies **Disabled** or *n minutes*, or *n hours* or **Custom** and a number of minutes. When not set to **Disabled**, the action locks the workstation after the specified amount of time elapses.

About the Windows File action

A Windows File action can verify the presence of specific files and can verify one or more file properties in situations where doing so increases confidence in the security of the client system. If a file with the described properties exists, the access policy passes the client to the successful branch. If the file does not exist, or a file exists but one or more properties are not correct, the access policy passes the client to the fallback branch.

The Windows File action provides the following configuration elements and options:

Continuously check the result and end the session if it changes

Specifies **Enabled** or **Disabled**.

FileName

Specifies the file name for which to check; for example, `notepad.exe` can be used to check for Windows Notepad.

MD5

Specifies the MD5 checksum. An MD5 checksum provides easily computable verification of the identity of a file using a cryptographic hash algorithm. The MD5 checksum is a 32-digit hexadecimal value. For example, the checksum for a zero-byte file is always `d41d8cd98f00b204e9800998ecf8427e`.

Size

Specifies the size of the file in bytes. The default value is 0 which is the same as not specifying a size; a size of zero (0) is not verified.

***Note:** A zero-byte file is specified with the MD5 checksum for a zero-byte file in the **MD5** field.*

Signer

Specifies the signer for the file. This can be left blank to omit checking for a signer.

Date

Specifies the file last modified date.

***Note:** The date must be translated first to GMT, and then to a 24-hour clock.*

Version

Specifies the version of the file. This can be left blank to omit checking for a version.

Version Comparison

Specifies the version comparison operator:

- = Specifies that the version to check for is the exact version specified in the **Version** field.
- < Specifies that the version to check for is a higher number than the version number specified in the **Version** field.
- > Specifies that the version to check for is a lower number than the version number specified in the **Version** field.

About Windows Health Agent

The Windows Health Agent action checks for health agent software on Windows-based client systems. When this action includes checks for multiple health agent types, if one specified type matches the software on the client system, the action passes, regardless of other health agent conditions that are specified in the action.

A Windows Health Agent action provides these settings and options:

Continuously check the result and end the session if it changes

Specifies **Enabled** or **Disabled**.

Vendor ID

Specifies a vendor ID (from the list of supported vendors) or **Any**.

Product ID

Specifies a product ID (from the list of supported products) or **Any**.

Version

Specifies a version; when specified, the Windows Health Agent action verifies the version of the software.

Policy Compliance

Specifies one of these values:

- **Enabled** - when selected, the action verifies that the client is compliant with the health policy specified by the site administrator.
- **Disabled** - when selected, the agent verifies that the client is out of compliance with the health policy specified by the site administrator.
- **Unspecified** - when selected, that action does not check for policy compliance.

About Windows Info

The Windows Info action determines whether the client uses particular versions of the Windows operating system and has applied specific patches or updates to Windows. The Windows Info action supplies several default branch rules for various Windows operating system versions or Windows operating system version and service pack combinations.

The Windows Info action supplies these conditions for defining branch rules.

Windows platform is

Specifies a platform; supported platforms are available for selection on a list.

Windows patch *n* is installed

Specifies a patch version or service pack number, such as SP1.

About Windows Process

The Windows Process action can verify that one or more particular processes are or are not running on a client system.

The Windows Process action provides these configuration elements and options:

Continuously check the result and end the session if it changes

Specifies **Enabled** or **Disabled**.

Expression

Specifies a Boolean expression to use to check for a process. The expression can include these wildcards: * and ?, and parentheses () to combine values, and the logical operators AND, OR, and NOT. This is the syntax for a process check expression: "*process name*" | (EXPRESSION) | NOT EXPRESSION | EXPRESSION AND EXPRESSION | EXPRESSION OR EXPRESSION

***Note:** Double quotes (" ") are required around each process name.*

Here is an example expression: ("winlogon.exe" AND "GoogleDesktop.exe") AND NOT "gator*". The expression checks running Windows processes for the presence of the winlogon.exe and GoogleDesktop.exe processes and the absence of any process with gator in the name.

About Windows Protected Workspace

The Windows Protected Workspace action configures a temporary Windows user workspace for a session. This workspace contains temporary Desktop and My Documents folders. The protected workspace control deletes the temporary workspace and all of the folder contents at the end of the session.

***Note:** The Windows Protected Workspace and the Windows Cache and Session Control actions are not compatible and should not be used in the same session.*

Close Google Desktop Search

Specifies whether to close Google Desktop Search before starting protected workspace.

Allow user to temporarily switch from Protected Workspace

Specifies whether a user can switch from the protected workspace. When set to **Enabled**, the action provides a link so that the user can temporarily switch from the protected workspace.

Allow user to use printers

Specifies whether a user can use printers.

Allow write access to USB flash drives

Specifies whether a user can write from the protected workspace to USB flash drives:

- **Disabled** does not allow users to write to any USB flash drives from the protected workspace.
- **All USB flash drives** allows a user to write to any USB flash drive from the protected workspace.
- **Only IronKey Secure Flash Drives** allows a user to write only to specialized, highly secured flash drives created by IronKey, Inc., from the protected workspace.

Allow user to burn CDs

Specifies whether a user can burn CDs from within the protected workspace.

Allow user to choose storage location

Specifies whether a user can choose the storage location for protected workspace files:

- **Enabled** allows users to select a storage location.
- **Disabled** stores files in the user's Document and Settings directory.

Enable persistent storage

Specifies whether data is saved on the system after the Protected Workspace session is closed:

- **Enabled** allows users to save encrypted data from the Protected Workspace session on the local system after the session exits. The files are automatically decrypted and available in the next Protected Workspace session.
- **Disabled** prevents users from storing Protected Workspace data in persistent storage.

Password protect new storage

Specifies whether the protected workspace requires a password to access data in persistent storage.

- **Enabled** requires the user to set a password to access persistent storage data.
- **Disabled** uses the default encryption and decryption, which is based on the server group name and storage device volume serial number.

Server group name

Specifies a group name for the server. This name is arbitrary, but limits persistent storage to that group name. For example, if a user connects to a protected workspace on a server with group name GroupA, and persistent storage is enabled, the user data is available when reconnecting to a server with the group name GroupA. However, if the user then connects to a server with persistent storage enabled, and the server group name GroupB, persistent data from the GroupA Protected Workspace session is not available in the new session, and a new persistent storage is defined.

About Windows Registry

The Windows Registry action verifies the existence or absence of certain keys and values in the Windows system registry database based on user-entered key values or Boolean expressions.

The Windows Registry action provides this configuration element:

Expression

Specifies a Boolean expression.

This is the syntax for registry checker expressions:

```
"key" comparison_operator data
```

```
"key" ISPR
```

```
"key"."value" comparison_operator data
```

```
"key"."value" ISPR
```

“key”

Represents a path in the Windows registry. Quotation marks are required around the path. If quotation marks exist as part of the registry path, they should be doubled (requires two sets of quotation marks).

“value”

Represents the name of the value. Quotation marks are required. If quotation marks exist as part of the value name, they should be doubled (requires two sets of quotation marks).

comparison_operator

Represents a comparison operator (< <= > >= =) or ISPR. ISPR verifies that a key or value is present. The equal sign (=) specifies equality.

Note: The operator == is not valid here.

data

Represents the content to compare against. Any spaces, commas, slashes, tabs, or other delimiters in the data must be enclosed in quotation marks. Data is interpreted as a version number when formatted like this: *d.d[.d][.d]*, *d, d[, d] [, d]* (where *d* is a number). Data is interpreted as a date when formatted like this: *mm/dd/yyyy*.

Table 2: Example expressions

Expression	Description
"HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\XP"	Checks for the presence of the specified path in the registry.
"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\InternetExplorer"."Version">="6.0.2900.2180"	Checks that the Internet Explorer version is greater than or equal to the value specified.

Expression	Description
<pre>"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\InternetExplorer"."Version">= "5.0.2800.0" AND "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\InternetExplorer"."Version" >= "6.0.2900.0"</pre>	Checks for the presence of Internet Explorer. With this registry check, the Internet Explorer version must be greater than or equal to 5.0.2800.0, and less than or equal to 6.0.2900.0.

About 32-bit registry keys on a 64-bit Windows client

On 64-bit Windows systems, the Windows Registry action can check for registry keys in the 64-bit registry or the 32-bit registry. The following registry root key names are supported:

- HKEY_CURRENT_USER
- HKEY_CURRENT_USER32
- HKEY_CURRENT_USER64
- HKEY_LOCAL_MACHINE
- HKEY_LOCAL_MACHINE32
- HKEY_LOCAL_MACHINE64
- HKEY_CLASSES_ROOT
- HKEY_CLASSES_ROOT32
- HKEY_CLASSES_ROOT64
- HKEY_USERS
- HKEY_USERS32
- HKEY_USERS64

An HKEY value specified with a 32 can provide a 32-bit view of a 64-bit registry. This is the perspective used by 32-bit applications running on a 64-bit operating system. An HKEY value specified with a 64 can provide a 64-bit view of the registry. This is the perspective used by native 64-bit applications.

Keys without a bit value specified use the default Windows registry redirectors, as specified by Microsoft. On a 32-bit Windows system, the number of bits specified in a registry key name is ignored.

About endpoint security (server-side) access policy items

In endpoint security (server-side) actions, the server queries clients and makes policy decisions based on information that a client presents to the server. For example, the Client Type action presents a query to find out what type of client is connecting, and routes the client to the different policy branches based on the results of the query. Endpoint security (server-side) access policy items do not require installation of client components.

About Client for MS Exchange

The Client for MS Exchange action determines whether a client is using Microsoft Exchange or ActiveSync protocols. This action includes two default branches: Client for MS Exchange and fallback. The Client for MS Exchange branch indicates that the client uses the Microsoft Exchange or ActiveSync protocol. A client for Microsoft Exchange is not a typical web browser and Access Policy Manager® (APM®) has the following restrictions on Client for MS Exchange access policy branches.

Behavioral restrictions

- APM does not attempt to perform authentication retries.
- A logon page action automatically works in clientless mode. (The access policy must include a logon page action.)
- Except for the logon page, APM cannot provide responses that require additional user input.

Limited supported actions

Microsoft Exchange devices support only the following actions. Therefore, only these actions are supported on a Client for MS Exchange access policy branch.

- Authentication actions:
 - AD Auth
 - AD Query
 - Client Cert Inspection
 - HTTP Auth
 - LDAP Auth
 - LDAP Query
 - NTLM Auth
 - RADIUS Auth
 - RADIUS Accounting
 - RSA SecurID Authentication
- Endpoint security (server-side) actions:
 - Client-Side Capability
 - Client OS
 - Landing URI
 - IP Geolocation Match

About Client OS

The Client OS action detects the operating system of the remote client. Access Policy Manager® detects this using information from the HTTP header. The action provides separate branches for separate operating systems. This action can be very useful at the beginning of an access policy. Each branch can include actions that are specific to a client operating system.

This figure shows the Client OS action and default branches, configured to allow access to clients on the Windows RT operating system and to deny access to all others.

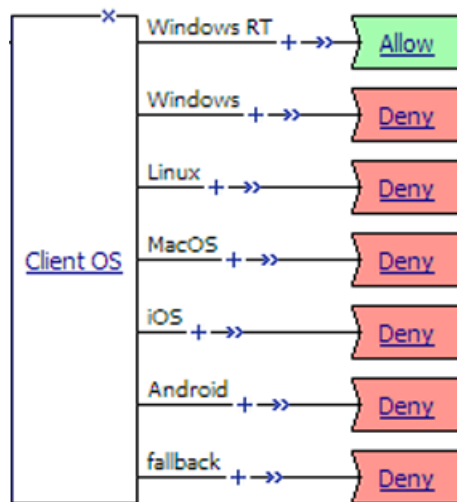


Figure 4: Client OS item with Allow ending configured on Windows RT branch

***Note:** In practice, actions would be specified on the access policy branches and might include logon actions, authentication actions, and other actions.*

About Client Type

The Client Type action determines whether the client is using a full browser, the BIG-IP® Edge Client, or another client to access the Access Policy Manager® (APM®). This action makes it possible to specify different actions for different client types in one access policy and, as a result, to use one virtual server for traffic from different client types. This figure shows the Client Type action as it looks when first added to an access policy.

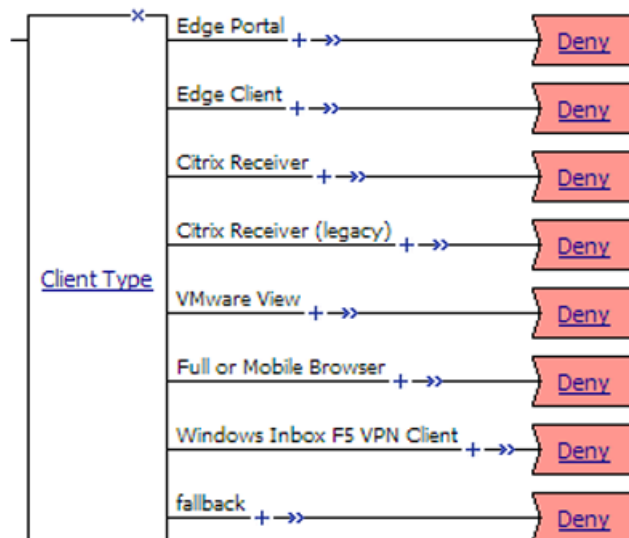


Figure 5: Client Type

By default, the Client Type action includes these branches:

Edge Portal

Indicates that the user is connecting with the BIG-IP® Edge Portal® mobile app.

Edge Client

Indicates that the user is connecting with the BIG-IP® Edge Client® or BIG-IP Edge Client app, supported on multiple devices and operating systems.

Citrix Receiver

Indicates that the user is connecting using a later Citrix Receiver client.

Citrix Receiver (legacy)

Indicates that the user is connecting using an earlier Citrix Receiver client (identified with PN Agent).

VMware View

Indicates that the user is connecting using a VMware Horizon View client.

Full or Mobile Browser

Indicates the user is connecting with a Windows web browser or a mobile browser.

Windows Inbox F5 VPN Client

Indicates the user is connecting using the Windows Inbox F5 VPN client.

fallback

Indicates the user is connecting with another method.

APM supports the client types on multiple operating systems. Refer to AskF5™ (support.f5.com) to look up the supported operating systems and versions in the compatibility matrix for your version of APM.

Note: To create additional branching for a client type based on operating system, you can add a client operating system (Client OS) action on the client type branch.

About Client-Side Capability

The Client-Side Capability action determines whether the client is fully capable of running endpoint security (client-side) actions. The Client-Side Capability action includes two branches.

Branch	Description
Full	Indicates that the user is connecting with a client that has full client-side check support.
fallback	Indicates that the user is connecting with a client that does not fully support client-side checks.

This action can be very useful as one of the first checks in an access policy. The **Full** branch can include the required client-side checks for those clients that are capable, while the fallback branch can lead to access policy branches for other clients.

This figure shows an example in which the Client-Side Capability action is used to verify that the client is capable of running a client-side check before running the client-side check for anti-spyware software.

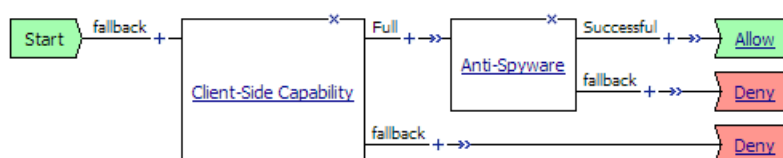


Figure 6: Client-side capability check before a client-side check

***Note:** In practice, an access policy would usually include a logon action, an authentication action, and other actions.*

About the Date Time action

The Date Time action checks the date or the time to support date- and time-based access. The Date Time action provides two default branch rules:

Weekend

Defined as Saturday and Sunday.

Business Hours

Defined as 8:00am to 5:00pm.

The Date Time action provides these conditions for defining branch rules.

Time From

Specifies a time of day. The condition is true at or after the specified time.

Time To

Specifies a time of day. This condition is true before or at the specified time.

Date From

Specifies a date. This condition is true at or after the specified date.

Date To

Specifies a date. This condition is true before or at the specified date.

Day of Week

Specifies a day. The condition is true for the entire day (local time zone).

Day of Month

Specifies the numeric day of month. This condition is true for this day every month (local time zone).

About IP Geolocation Match

The IP Geolocation Match action determines a user's physical location by comparing the user's IP address to an internal database. The IP Geolocation Match action can make a match based on one or more location parameters.

The default branch rule is **IP Geolocation Country code is** US.

The IP Geolocation Match action provides these conditions for defining branch rules.

IP Geolocation Continent code is

Specifies that the user's IP address must match the specified continent code.

IP Geolocation Country code is

Specifies that the user's IP address must match the specified country code.

IP Geolocation Country name is

Specifies that the user's IP address must match the specified country name.

IP Geolocation State/Region is

Specifies that the user's IP address must match the specified region or state.

About IP Reputation

When an IP Reputation action is included in an access policy, Access Policy Manager® (APM®) searches for the IP address in the IP intelligence database. The IP intelligence database contains only IP addresses that are considered untrustworthy, along with a category for each that describes why it is not trusted.

APM provides these default branch rules for the IP Reputation action.

Bad

The IP address exists in the IP intelligence database. The expression for this branch rule includes every IP reputation category. For example, the rule includes expressions such as IP Reputation is: Spam Sources OR IP Reputation is: Proxy, and so on. If any IP reputation category is acceptable at your site, you should update this rule or create and use another rule.

Good

The IP address is not found in the IP intelligence database.

fallback

The IP intelligence database is inaccessible for some reason. This can be due to a misconfiguration or a problem with a license or Internet connectivity.

About IP Subnet Match

The IP Subnet Match action determines whether the client IP address matches an IP subnet. The IP Subnet Match action provides this configuration option:

IP Subnet Match - specifies a subnet, such as 10.0.0.0/8.

About Jailbroken or Rooted Device Detection

The Jailbroken or Rooted Device Detection action determines whether a mobile device is jailbroken or rooted. This action provides two default branches: Jailbroken or Rooted Device and fallback.

About Landing URI

The Landing URI action checks the landing URI with which the user accessed the access policy. The default Landing URI action includes two branches.

Branch	Description
Landing URI	Indicates that the user is connecting with a URI that matches a specified landing URI. Specifies <code>/uri1</code> or <code>/uri1/</code> as the default landing URI. To use this action, it is required to edit the branch rules to specify an actual landing URI.
fallback	Indicates that the user is connecting with a different landing URI.

This figure shows a branch rule that determines whether the address that the user typed includes the string `/owa` or `/owa/`, either of which is part of the typical landing URI for an Outlook Web Access connection.

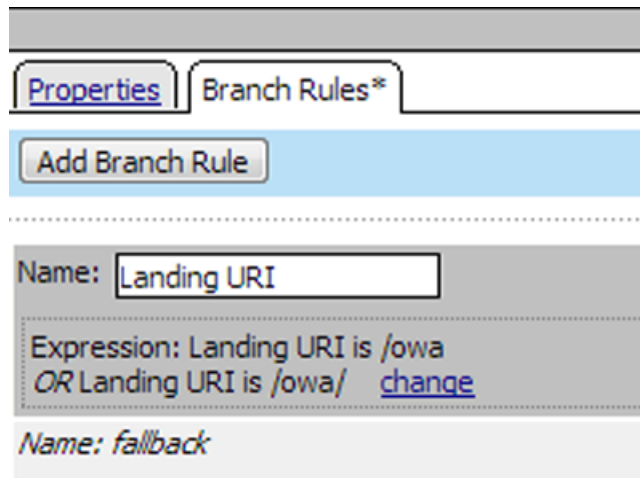


Figure 7: Landing URI branch rule with updated expression

About the License action

The License action provides the ability to create branch rules based on license use. It can check the number of remaining licenses against an absolute value or the percentage of licenses remaining against a threshold. A License action can check access licenses, connectivity licenses, and concurrent users.

The License action supplies this default branch rule: **Remaining Global Access license count is above percentage threshold: 20**. This branch rule can be deleted or changed. The License action supplies these conditions for configuring branch rules:

- **Remaining Global Access License count is above absolute value** - checks number of remaining global access licenses against the number that you specify.
- **Remaining Global Access License count is above percentage threshold** - checks percentage of global access licenses that remain against the threshold that you specify.
- **Remaining Global Connectivity License count is above absolute value** - checks number of remaining global connectivity licenses against the number that you specify.
- **Remaining Global Connectivity License count is above percentage threshold** - checks percentage of global connectivity licenses that remain against the threshold that you specify.
- **Remaining Concurrent User count is above absolute value** - checks number of remaining concurrent user licenses against the number that you specify.
- **Remaining Concurrent User count is above percentage threshold** - checks percentage of concurrent user licenses that remain against the threshold that you specify.

If the license check does not match the specified conditions, the access policy sends the user to the fallback branch.

About general purpose items

General purpose items can be used in any case and can be placed anywhere in an access policy. These items support:

- Logging a message and variables
- Sending email
- Displaying a message

- Processing an iRule
- Providing a choice between two options
- Running user-configured rules
- Reading from and writing to a local user database

When an administrator adds these items to an access policy, the administrator specifies the message (to log, to display, to email), any options that a user can choose, the iRule to process, and so on, to suit the situation.

About the Decision Box action

A Decision Box action presents two options to the user. These options are presented as link text, preceded by images.

A Decision Box action can be useful after a client fails an endpoint security check, or after a user fails to authenticate. When this occurs, a branch rule can provide an option to allow the user to continue onto a guest or quarantine network that provides only limited access to a segregated subnet. The other branch can provide an option to log out, and present the user with a logon denied ending.

The figure illustrates the configuration of a Decision Box action. The top part is a flowchart showing the logic: an 'Anti-Spyware' check leads to a 'Successful' path (allowing access) or a 'fallback' path to a 'Decision Box'. The 'Decision Box' offers two options: 'Guest net' (leading to 'Advanced Resource Assign' and then 'Allow') and another 'fallback' path (leading to 'Deny').

The bottom part is the configuration interface for the 'Decision Box' action. It includes a 'Properties*' tab and a 'Branch Rules' tab. The 'Name' is set to 'Decision Box'. Under the 'Customization' section, the 'Language' is set to 'en'. The 'Message' field contains: 'Your anti-spyware software is not up-to-date. Please choose one of these options:'. The 'Field 1 image' is set to 'green icon'. 'Option 1' is 'Log on to the Guest network.'. The 'Field 2 image' is set to 'red icon'. 'Option 2' is 'Log out'.

Figure 8: Configuring a decision box to appear after a failed endpoint security check (with decision box detail)

Another use of the Option 2 branch is to allow the user to continue to a redirect ending that takes the user to a helpful URL, for example, to the web site of an antivirus vendor to download virus database updates.

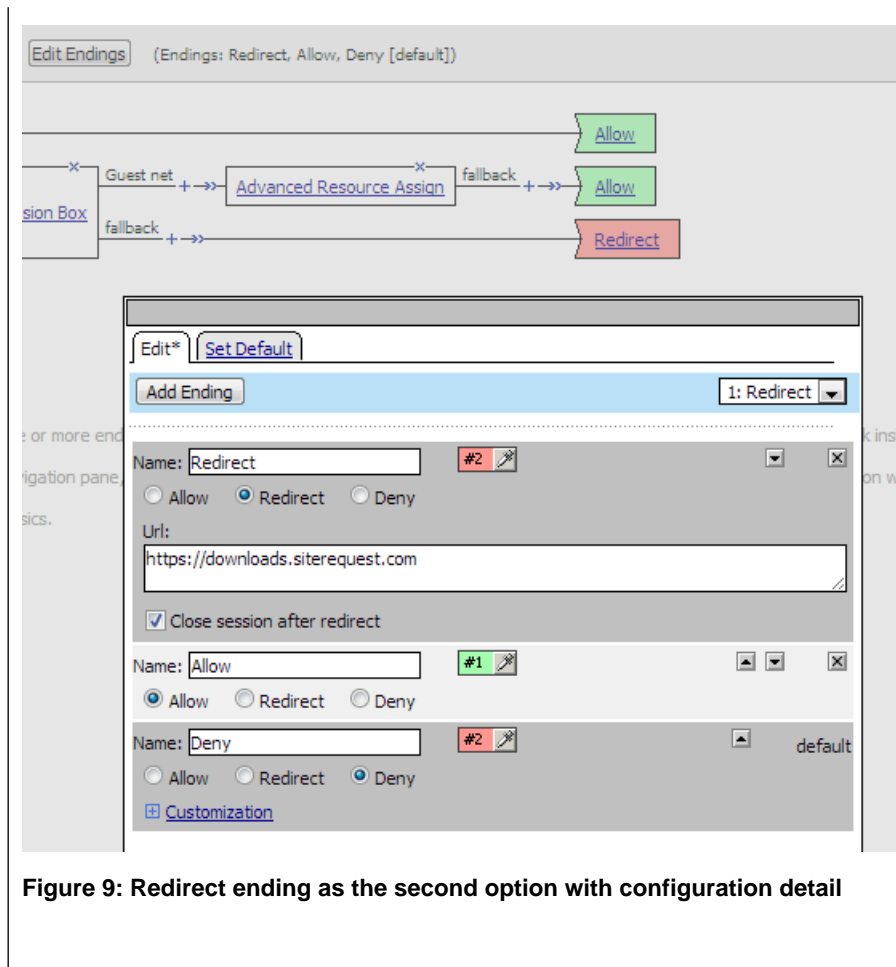


Figure 9: Redirect ending as the second option with configuration detail

About the Email action

An Email action can send email. An Email action provides these configuration options and elements:

SMTP Configuration

Specifies an SMTP configuration on the BIG-IP® system.

From

Specifies the sender which can be a string or a session variable name or both. For example:

`APM@vs-#{session.server.network.name}`

To

Specifies the recipient. This can be a fully qualified email address or a session variable name; for example: `%{session.ad.last.attr.mail}`

CC

Specifies recipients to be copied on the mail. This can be fully qualified email addresses or session variable names.

Subject

Specifies the subject of the email message. This can be a string, a session variable name, or a combination of strings and session variable names.

Message

Specifies the message to send. This can be a string, a session variable name, or a combination of strings and session variable names.

About the Empty action

An Empty action has no explicit configuration. The action allows a user to create rules only, using the Branch Rules tab.

About iRule Event

An iRule Event action adds iRule processing to an access policy at a specific point. An iRule Event provides one configuration option: ID, which specifies an iRule event ID.

Note: *iRule event access policy items must be processed and completed before the access policy can continue.*

An iRule Event action can occur anywhere in an access policy.

About Local Database

The Local Database action can read and write information about a user in a local user database.

Note: *Changes that an administrator makes to a local user database, whether from the Configuration utility or the command line, can override the changes that this action makes.*

A Local Database action provides the following configuration elements and options:

LocalDB Instance

Specifies a local user database instance from a list.

User Name

Specifies a user name from a list.

Note: *The same user name can exist in more than one local user database.*

Allow User Create

Specifies whether to create a user dynamically when trying to write information for a user that is not in the database already.

Note: *Dynamically created users exist temporarily and are regularly purged from the database. Static users, created by an administrator using the Configuration Utility or the command line, are not purged.*

Add new entry

Specify actions that read from and write to specific database properties. An entry includes these elements:

- **Action** - specifies **Read** or **Write**.
- **Destination** - specifies where to store the value that is being read or written. For the **Read** action, this specifies a variable. The value read from the database is stored in this variable. For the **Write** action, this specifies a DB property (selectable from a list). The value of an expression is stored in this DB property. The **DB Property** list includes these items:

- **locked_out** - a number; when 0, the user is not locked out. When greater than 0, the user is locked out.
- **login_failures** - a number; the number of login failures currently recorded for the user.
- **groups** - text; names of membership groups specified for the user in the local user database.

***Note:** Groups specified in the local user database are not verified against external systems.*

Source

Specifies where to get the value to read or to write. For **Read**, specifies a database property (selectable from a list). The value in the database property is read. For **Write**, specifies an expression. The value of the expression is written to the database to the database property.

About the Logging action

The Logging action adds logging for session variables to the access policy. This action is useful for tracing the session variables that are created for a specific category, or in a specific branch.

***Note:** A session variable might or might not exist at the time of logging; depending on the result of the access policy branch, or results of processing the access policy.*

The Logging action provides these configuration elements and options:

Log Message

Specifies text to add to the log file.

Session Variables

Specifies a session variable from a list of predefined session variables or a custom session variable.

About the Message Box action

A Message Box action presents a message to the user, and prompts the user to click a link to continue. The message box has no effect on the user's access to the network or the preceding or following access policy checks. A message box can be used, for example, to warn a user about a redirect to a guest network, or that the client certificate failed to authenticate, or to display a message about the results of a rule branch in the access policy.

A Message Box action provides these configuration elements and options:

Language

Specifies the language to use to customize this logon page. When a user selects a language, the content in the remaining fields display in the selected language.

***Note:** Languages on the list reflect those that are configured in the access profile.*

Message

Specifies the message to present to the user.

Link

Specifies the message that appears as the link text.

About authentication items

Authentication items perform authentication or authentication-related functions, such as:

- Verify credentials (or a PIN or a token)
- Inspect SSL certificates
- Check SSL certificate revocation status
- Verify the result of passwordless authentication
- Perform accounting, and so on.

An authentication item usually follows a logon item or another authentication item in an access policy. An access policy can contain any number of authentication items.

An administrator that configures authentication items can make these choices:

- Specify an AAA server (or pool in cases where high availability is supported) against which to authenticate. Access Policy Manager® (APM®) supports many types of AAA servers.
- Inspect the SSL certificate presented during the initial SSL handshake, or specify on-demand certificate authentication (to re-negotiate the SSL connection). On-demand authentication is not supported in every type of access configuration.
- Select a Certificate Revocation Location (CRL) or Online Certificate Status Protocol (OCSP) responder for verifying revocation status.

Note: Other configuration objects must be created before configuring an authentication item or before a particular type of authentication is fully configured and working. Refer to *BIG-IP Access Policy Manager: Authentication and Single Sign-On on the AskF5™ web site at <http://support.f5.com/kb/en-us.html>*.

About AD Auth

An AD Auth action authenticates a user against an AAA Active Directory server. In an access policy, an authentication action typically follows a logon action that collects credentials.

An AD Auth action provides these configuration elements and options:

Type

Specifies Authentication, the type of this Active Directory action.

Server

Specifies an Active Directory server; servers are defined in the Access Policy AAA servers area of the Configuration utility.

Cross Domain Support

Specifies whether AD cross domain authentication support is enabled for this action.

Complexity check for Password Reset

Specifies whether Access Policy Manager® (APM®) performs a password policy check. APM supports these Active Directory password policies:

- Maximum password age
- Minimum password age
- Minimum password length
- Password must meet complexity requirements

APM must retrieve all related password policies from the domain to make the appropriate checks on the new password.

***Note:** Because this option might require administrative privileges, the administrator name and password might be required on the AAA Active Directory server configuration page.*

***Note:** Enabling this option increases overall authentication traffic significantly because APM must retrieve password policies using LDAP protocol and must retrieve user information during the authentication process to properly check the new password.*

Show Extended Error

When enabled, causes comprehensive error messages generated by the authentication server to display on the user's logon page. This setting is intended only for use in testing, in a production or debugging environment. If enabled in a live environment, your system might be vulnerable to malicious attacks. (When disabled, displays non-comprehensive error messages generated by the authentication server on the user's logon page.)

Max Logon Attempts Allowed

Specifies the number of user authentication logon attempts to allow. A complete logon and password challenge and response is considered as one attempt.

Max Password Reset Attempts Allowed

Specifies the number of times that APM allows the user to try to reset password.

About AD Query

An AD Query action performs a query against an AAA Active Directory server. An AD Query action provides these configuration elements and options:

Type

Specifies Query, the type of this Active Directory action.

Server

Specifies an Active Directory server; servers are defined in the Access Policy AAA servers area of the Configuration utility.

SearchFilter

Specifies the search criteria to use when querying the Active Directory server for the user's information. Session variables are supported as part of the search query string.

Fetch Primary Group

Specifies whether to retrieve a user's primary group Distinguished Name for use in the access policy.

Cross Domain Support

Specifies whether AD cross domain authentication support is enabled for this action.

Fetch Nested Groups

When disabled, associates the user only to the groups to which they belong directly. When enabled, associates the user to all groups that are nested under the groups that they directly belong to. For example, if the user belongs to Group 1 and Group 2, and Group 1 is a member of Group 3 and Group 4, enabling this setting allows the user to obtain privileges from all groups.

Complexity check for Password Reset

Specifies whether Access Policy Manager® (APM®) performs a password policy check. APM supports these Active Directory password policies:

- Maximum password age
- Minimum password age
- Minimum password length
- Password must meet complexity requirements

APM must retrieve all related password policies from the domain to make the appropriate checks on the new password.

Note: Because this option might require administrative privileges, the administrator name and password might be required on the AAA Active Directory server configuration page.

Note: Enabling this option increases overall authentication traffic significantly because APM must retrieve password policies using LDAP protocol and must retrieve user information during the authentication process to properly check the new password.

Max Password Reset Attempts Allowed

Specifies the number of times that APM allows the user to try to reset password.

Prompt user to change password before expiration

Specifies whether to warn the user at a set time before the password expires and provide the option to change the password.

About Client Cert Inspection

Normally, when a client makes an HTTPS request, an SSL handshake request occurs at the start of an SSL session. If the connection is allowed, the Client Cert Inspection action can check the result of the request.

The Client Cert Inspection action provides two branches: Successful and fallback.

About CRLDP Auth

A CRLDP Auth action retrieves a Certificate Revocation List (CRL) from a network location (*distribution point*). A distribution point is either an LDAP Uniform Resource Identifier (URI), a directory path that identifies the location where the CRLs are published, or a fully qualified HTTP URL. An CRLDP Auth action provides these configuration elements and options:

CRLDP Server

Specifies a CRLDP server; servers are defined in the Access Policy AAA servers area of the Configuration utility.

About HTTP Auth

A HTTP Auth action authenticates a user against an HTTP AAA server. An HTTP Auth action provides these configuration elements and options:

AAA Server

Specifies an HTTP AAA server; servers are defined in the Access Policy AAA servers area of the Configuration utility.

About Kerberos Auth

A Kerberos Auth action retrieves user credentials using a Kerberos ticket.

Note: *In an access policy, an HTTP 401 Response action typically precedes a Kerberos Auth action.*

A Kerberos Auth action provides these configuration elements and options:

AAA Server

Specifies a Kerberos server; servers are defined in the Access Policy AAA servers area of the Configuration utility.

Request Based Auth

Specifies whether per request based authentication is enabled. When disabled, authentication occurs only while executing the access policy.

Max Logon Attempts Allowed

Specifies the number of user authentication logon attempts to allow. A complete logon and password challenge and response is considered as one attempt.

About LDAP Query

An AD Query action performs a query against an AAA LDAP server. An AD Query action provides these configuration elements and options:

Type

Specifies Query, the type of this LDAP action.

Server

Specifies an LDAP server; servers are defined in the Access Policy AAA servers area of the Configuration utility.

SearchDN

Specifies the base node of the LDAP server search tree to start the search with.

SearchFilter

Specifies the search criteria to use when querying the LDAP server for the user's information. Session variables are supported as part of the search query string. When strings are used, they must be enclosed in parentheses; for example, (`sAmAccountName=%{session.logon.last.username}`).

Fetch Nested Groups

When disabled, associates the user to the groups that they directly belong to. When enabled, associates the user to all groups that are nested under the groups that they directly belong to. For example, if the user belongs to Group 1 and Group 2, and Group 1 is a member of Group 3 and Group 4, enabling this setting allows the user to obtain privileges from all groups.

Required Attributes (optional)

By default, the server loads all user attributes if no required attributes are specified. However, system performance can improve if fewer attributes are returned.

About LocalDB Auth

The LocalDB Auth action can authenticate a user against a local user database instance. The LocalDB Auth action can lock a user out of a local user database instance if they fail to log on within a specified number of attempts.

Note: For enhanced security, typically, Local Database actions should be placed before and after a LocalDB Auth action to read and write user information to track non-static users (those not created by an administrator) that attempt repeatedly to logon and fail.

A LocalDB Auth action provides these configuration elements and options.

LocalDB Instance

Specifies a local user database instance.

Max Logon Attempts Allowed

A number from 1 to 5.

About NTLM Auth Result

If NTLM authentication occurs, it happens before the access policy runs. The NTLM Auth Result action checks the result and provides two branches: Successful and fallback.

About OAM authentication

An OAM action authenticates a user against an Oracle Access Manager (OAM) server. An OAM action provides these configuration elements and options:

Server

Specifies the OAM server. (Servers are defined in the Access Policy AAA servers area of the Configuration utility.)

URL

Specifies a URL resource, for example, `http://plum.tree.lab2.sp.companynet.com/`. This resource must respond with a challenge to a non-authenticated request.

Agent Action

Specifies the type for the OAM action.

- **Authentication only** - Specifies that this action performs OAM authentication.
- **Authentication and Authorization** - Specifies that this action performs both OAM authentication and authorization.

Show Extended Error

When enabled, causes comprehensive error messages generated by the authentication server to display on the user's logon page. This setting is intended only for use in testing, in a production or debugging environment. If enabled in a live environment, your system might be vulnerable to malicious attacks. (When disabled, displays non-comprehensive error messages generated by the authentication server on the user's logon page.)

Max Logon Attempts Allowed

Specifies the number of user authentication logon attempts to allow. A complete logon and password challenge and response is considered as one attempt.

About OCSP Auth

An OCSP Auth action retrieves the revocation status of an X.509 certificate by sending the certificate information to a remote Online Certificate Status Protocol (OCSP) responder. An OCSP Auth action typically follows an action that makes certificate information available, either a Client Cert Inspection or On-Demand Cert Auth action.

An OCSP Auth action provides these configuration elements and options:

OCSP Responder

Specifies the OCSP Responder server; servers are defined in the Access Policy AAA servers area of the Configuration utility.

About On-Demand Cert Auth

Typically, when a client makes an HTTPS request, an SSL handshake request occurs at the start of an SSL session. If the client SSL profile skips the initial SSL handshake, an On-Demand Cert Auth action can re-negotiate the SSL connection from an access policy by sending a certificate request to the user. This prompts a certificate screen to open. After the user provides a valid certificate, the On-Demand Cert Auth action checks the result of certificate authentication. The agent verifies the value of the session variable `session.ssl.cert.valid` to determine whether authentication was a success.

The On-Demand Cert Auth action provides one configuration option, **Auth Mode**, with two supported modes:

Request

With this mode, the system requests a valid certificate from the client, but the connection does not terminate if the client does not provide a valid certificate. Instead, this action takes the fallback route in the access policy. This is the default option.

Require

With this mode, the system requires that a client provides a valid certificate. If the client does not provide a valid certificate, the connection terminates and the client browser stops responding.

***Note:** For an iPod or an iPhone, the **Require** setting must be used for On-Demand certificate authentication. To pass a certificate check using Safari, the user is asked to select the certificate multiple times. This is expected behavior.*

About OTP Generate

The OTP Generate action can generate a one-time use time-limited password. This action does not send the one-time password to a user. Typically, an OTP Generate action precedes other actions that send the password (the Email action, for example) and then verify it (OTP Verify action). The OTP Generate action provides these configuration options:

OTP length

Specifies the length of the one-time password. Defaults to 6.

OTP timeout

Specifies the number of seconds that the password is valid. Defaults to 300.

About OTP Verify

In an access policy, the OTP Verify action checks for a match between a user-entered password and the one-time password generated previously by the OTP Generate action. The OTP Verify action also verifies that the one-time password has not expired. The OTP Verify action provides this configuration option:

Max Logon Attempts Allowed

Limits the number of logon attempts.

About SAML Auth

The SAML Auth action authenticates against an external SAML Identity Provider (IdP). This action is for use when the BIG-IP® system is configured as a SAML service provider and supports connections initiated at SAML service providers.

The SAML Auth action provides this configuration element:

AAA server

Specifies an external SAML IdP.

***Note:** IdPs are specified in SAML IdP connector configurations.*

About RADIUS Acct

A RADIUS Acct action reports user session information to an external RADIUS accounting server; it does not perform authentication.

A RADIUS Acct action provides these configuration elements and options:

AAA Server

Specifies the RADIUS server; servers are defined in the Access Policy AAA servers area of the Configuration utility.

About RADIUS Auth

A RADIUS Auth action authenticates a client against an external RADIUS server. A RADIUS Auth action provides these configuration elements and options:

AAA Server

Specifies the RADIUS accounting server; servers are defined in the Access Policy AAA servers area of the Configuration utility.

Show Extended Error

When enabled, causes comprehensive error messages generated by the authentication server to display on the user's logon page. This setting is intended only for use in testing, in a production or debugging environment. If enabled in a live environment, your system might be vulnerable to malicious attacks.

(When disabled, displays non-comprehensive error messages generated by the authentication server on the user's logon page.)

Max Logon Attempts Allowed

Specifies the number of user authentication logon attempts to allow. A complete logon and password challenge and response is considered as one attempt.

About RSA SecurID

An RSA SecurID action authenticates a user name and PIN code or token against a SecurID server. In an access policy, an authentication action typically follows a logon action that collects credentials. An RSA SecurID action provides these configuration elements and options:

AAA Server

Specifies the RSA SecurID server; servers are defined in the Access Policy AAA servers area of the Configuration utility.

Show Extended Error

When enabled, causes comprehensive error messages generated by the authentication server to display on the user's logon page. This setting is intended only for use in testing, in a production or debugging environment. If enabled in a live environment, your system might be vulnerable to malicious attacks. (When disabled, displays non-comprehensive error messages generated by the authentication server on the user's logon page.)

Max Logon Attempts Allowed

Specifies the number of user authentication logon attempts to allow. A complete logon and password challenge and response is considered as one attempt.

About TACACS+ Acct

A TACACS+ Acct action adds Terminal Access Controller Access Control System (TACACS+) accounting to an access policy. The accounting service sends `start` and `stop` accounting records to the remote server.

A TACACS+ Acct action provides these configuration elements and options:

AAA Server

Specifies the TACACS+ accounting server; servers are defined in the Access Policy AAA servers area of the Configuration utility.

About TACACS+ Auth

A TACACS+ Acct action authenticates a user against a Terminal Access Controller Access Control System (TACACS+) server. In an access policy, an authentication action typically follows a logon action that collects credentials. A TACACS+ Acct action provides these configuration elements and options:

AAA Server

Specifies the TACACS+ accounting server; servers are defined in the Access Policy AAA servers area of the Configuration utility.

Max Logon Attempts Allowed

Specifies the number of user authentication logon attempts to allow. A complete logon and password challenge and response is considered as one attempt.

About Transparent Identity Import

A Transparent Identity Import action obtains an IP-address-to-username-mapping, if it exists, from an IF-MAP server located on the BIG-IP[®] system. If the mapping exists, the user identity is assumed to be known.

Note: *An IF-MAP server exists and is populated when the F5[®] DC Agent is installed, configured, and operating correctly in your network.*

A Transparent Identity Import action provides two branches: Associated and fallback.

Chapter 4

Session Variables

- *About session variables*
 - *About session variable names*
 - *Session variables reference*
-

About session variables

An access policy stores the values that actions return in session variables. A *session variable* contains a number or string that represents a specific piece of information. This information is organized in a hierarchical arrangement and is stored as the user's session data.

The Current Sessions report in the Access Policy Manager® Reports area displays all session variables for a session. Session variables can be useful in access policies to achieve various results, including:

- Customizing access rules or defining your own access policy rules.
- Providing different outcomes for policies based on the values in the session variables.
- Determining which resources to assign to users (with the Resource Assign action).

About session variable names

The name of a session variable consists of multiple hierarchical nodes separated by periods (.).

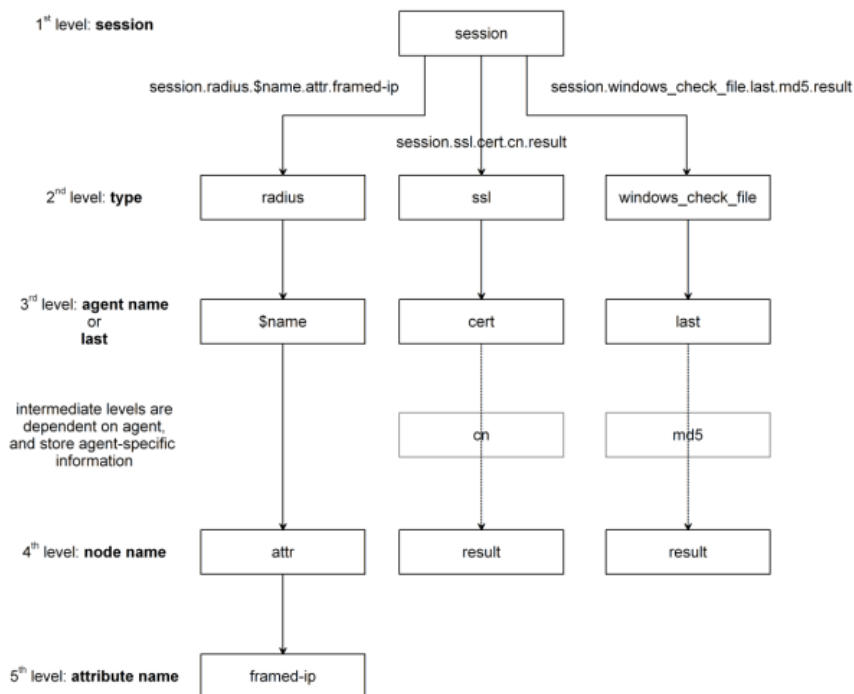


Figure 10: How APM constructs session variable names

Session variables for Active Directory authentication and query

Access Policy Manager® names session variables in the following manner:

- session.ad.<username>.queryresult = query result (0 = failed, 1=passed)
- session.ad.<username>.authresult = authentication result (0 = failed, 1=passed)

- `session.ad.<username>.attr.<attr_name>` = the name of an attribute retrieved during the Active Directory query. Each retrieved attribute is converted to a separate session variable.

Note that attributes assigned to a user on the AAA server are specific to that server, and not to Access Policy Manager.

Session variables reference

This table includes session variables and related reference information.

Session variables for access policy action items

Action Item	Session Variable	Type	Description
Denied Ending	<code>session.policy.result</code>	string	Access policy result: the access policy ended at Deny. The value is "access_denied".
Redirect Ending	<code>session.policy.result</code>	string	Access policy result: the access policy ended at Redirect. The value is "redirect".
	<code>session.policy.result.redirect.url</code>	string	URL specified in the redirect, for example, "http://www.siterequest.com"
Allowed Ending	<code>session.policy.result</code>	string	Access policy result: the access policy ended at Allow. The value is "allowed".
	<code>session.policy.result.webtop.network_access.autolaunch</code>	string	Name of the resource that is automatically started for a network access webtop
	<code>session.policy.result.webtop.type</code>	string	Type of webtop resource: "network_access" or "web_application".
Session management	<code>session.ui.mode</code>	enum	UI mode, as determined by HTTP headers.
	<code>session.ui.lang</code>	string	Language in use in the session, for example "en" (English).
	<code>session.ui.charset</code>	string	Character set used in the session.
	<code>session.client.type</code>	enum	Client type as determined by HTTP headers: portalclient "Standalone"
	<code>session.client.version</code>	string	
	<code>session.client.js</code>	bool	
	<code>session.client.activex</code>	bool	
	<code>session.client.plugin</code>	bool	

Action Item	Session Variable	Type	Description
	<code>session.client.platform</code>	string	Client platform as determined by HTTP headers: <ul style="list-style-type: none"> "WinNT" "Win2k" "WinXP" "WinVI" "Linux" "MacOS" "iOS" "Android"
	<code>session.user.access_mode</code>	string	Enables direct access to a Citrix resource from the webtop. Example: "local"
Active Directory action	<code>session.ad.\$name.queryresult</code>	bool	0 or 1. <ul style="list-style-type: none"> 0 - Active Directory query failed 1 - Active Directory query passed
	<code>session.ad.\$name.authresult</code>	bool	0 or 1. <ul style="list-style-type: none"> 0 - Active Directory authentication failed 1 - Active Directory authentication passed
	<code>session.ad.\$name.attr.\$attr_name</code>	string	Users attributes retrieved during Active Directory query. Each attribute is converted to a separate session variable.
	<code>session.ad.\$name.attr.group.\$attr_name</code>	string	User's group attributes retrieved during Active Directory query. Each group attribute is converted to a separate session variable.
Advanced Resource Assign	<code>session.assigned.bwc.dynamic</code>	string	Name of the assigned dynamic bandwidth control policy
	<code>session.assigned.bwc.static</code>	string	Name of the assigned static bandwidth control policy
Client certificate authentication	<code>session.ssl.cert.x509extension</code>	string	X509 extensions
	<code>session.ssl.cert.valid</code>	string	Certificate result: OK or error string
	<code>session.ssl.cert.exist</code>	integer	0 or 1. <ul style="list-style-type: none"> 0 - Certificate does not exist 1 - Certificate exists
	<code>session.ssl.cert.version</code>	string	Certificate version
	<code>session.ssl.cert.subject</code>	string	Certificate subject field
	<code>session.ssl.cert.serial</code>	string	Certificate serial number

Action Item	Session Variable	Type	Description
	session.ssl.cert.end	string	Validity end date
	session.ssl.cert.start	string	Validity start date
	session.ssl.cert.issuer	string	Certificate issuer
	session.ssl.cert.whole	string	The whole certificate
Decision box	session.decision_box.last.result	integer	0 or 1. <ul style="list-style-type: none"> 0 - User chooses option 2 on the decision page, which corresponds to the fallback rule branch in the action. 1 - User chooses option 1 on the decision page
File check	session.windows_check_file.\$name.item_0.exist	string	True - if all files exist on the client.
	session.windows_check_file.\$name.item_0.result	integer	Set when files on the client meet the configured attributes.
	session.windows_check_file.\$name.item_0.md5	string	MD5 value of a checked file.
	session.windows_check_file.\$name.item_0.version	string	Version of a checked file.
	session.windows_check_file.\$name.item_0.size	integer	File size, in bytes.
	session.windows_check_file.\$name.item_0.modified		Date the file was modified in UTC form.
	session.windows_check_file.\$name.item_0.signer		File signer information.
LDAP action	session.ldap.\$name.authresult	bool	0 or 1. <ul style="list-style-type: none"> 0 - LDAP authentication failed 1 - LDAP authentication passed
	session.ldap.\$name.attr.\$attr_name	string	Users attributes retrieved during LDAP query. Each attribute is converted to a separate session variable.
	session.ldap.\$name.queryresult	bool	0 or 1. <ul style="list-style-type: none"> 0 - LDAP query failed 1 - LDAP query passed
Logon Page (CAPTCHA challenge)	session.logon.captcha.tracking	unsigned integer	A bitmask used when CAPTCHA is enabled. <ul style="list-style-type: none"> Bit in 0 position - Track successful and unsuccessful logon attempts by IP address Bit in 1 position - Track successful and unsuccessful logon attempts by user name <hr/> <i>Note: Should not be used by external modules because it is intended for very specific purposes.</i> <hr/>

Action Item	Session Variable	Type	Description
Machine Cert Auth	<code>session.check_machinecert.last.result</code>	integer	<p>0, 1, 2, or -2.</p> <ul style="list-style-type: none"> 0 - Neither certificate nor private key found. 1 - Both certificate and private key found. 2 - Certificate found, but private key not found. -2 - Various errors, such as: Nothing received from client. Data received is not in correct format. Incorrect configuration. (For example, CA profile is not configured). Linux client is trying to access the agent. <hr/> <p>Note: The Machine Cert Auth action is not supported on Linux.</p> <hr/>
OTP Generate	<code>session.otp.assigned.val</code>	string	Generated one-time password value to send to the end user. Example message: One-Time Passcode: <code>{session.otp.assigned.val}</code>
	<code>session.otp.assigned.expire</code>	string	Internally used timestamp; OTP expiration in seconds since this date and time: (00:00:00 UTC, January 1, 1970)
	<code>session.otp.assigned.ttl</code>	string	OTP time-to-live; configurable as OTP timeout in seconds. Example message: OTP expires after use or in <code>{session.otp.assigned.ttl}</code> seconds
OTP Verify	<code>session.otp.verify.last.authresult</code>	bool	<p>0 or 1.</p> <ul style="list-style-type: none"> 0 - OTP authentication failed 1 - OTP authentication passed
RADIUS action	<code>session.radius.\$name.authresult</code>	bool	<p>0 or 1.</p> <ul style="list-style-type: none"> 0 - RADIUS authentication failed 1 - RADIUS authentication passed
	<code>session.radius.\$name.attr.\$attr_name</code>	string	User attributes retrieved during RADIUS authentication. Each attribute is converted to a separate session variable.
Resource allocation	<code>session.assigned.resources</code>	string	Space-delimited list of names of assigned resources.

Action Item	Session Variable	Type	Description
	<code>session.assigned.webtop</code>	string	Name of the assigned webtop.
Windows Info	<code>session.windows_info_os.\$name.ie_version</code>	string	Stores the Internet Explorer version
	<code>session.windows_info_os.\$name.ie_updates</code>	string	List of installed SP and KB fixes for Internet Explorer. For example: " SP2 KB12345 KB54321 "
	<code>session.windows_info_os.\$name.platform</code>	string	Platform. <ul style="list-style-type: none"> "Win7" - Windows 7 "Win8" - Windows 8 "WinVI" - Windows "WinXP" - Windows XP "Win2003" - Windows 2003 Server "WinLH" - Windows 2008
	<code>session.windows_info_os.\$name.updates</code>	string	List of installed SP and KB fixes for Windows. For example, " SP2 KB12345 KB54321 "
	<code>session.windows_info_os.\$name.user</code>	string	List of current Windows user names
	<code>session.windows_info_os.\$name.computer</code>	string	List of computer names
Windows Process	<code>session.windows_check_process.\$name.result</code>	integer	0, 1, or -1. <ul style="list-style-type: none"> 0 - Failure 1 - Success -1 - Invalid check expression
	<code>session.windows_check_registry.\$name.result</code>	integer	0, 1, or -1. <ul style="list-style-type: none"> 0 - Failure 1 - Success -1 - Invalid check expression

Chapter 5

Tcl Usage

- *About Tcl usage in APM*
 - *Tcl syntax notes*
 - *Tcl examples*
-

About Tcl usage in APM

The Tcl programming language can be used for writing advanced branch rules in the visual policy editor, and for assigning variables to custom expressions in the Variable Assign action.

Note: Use of Tcl is optional; it provides an alternative to using the expression builder (in the visual policy editor), and to using other options provided by the Variable Assign action.

Tcl syntax notes

Access Policy Manager® (APM®) supports standard Tcl syntax and additional commands and operators listed in the table.

Standard Tcl Syntax

APM supports the various facilities provided by the Tcl language; for example, loops (`while`, `foreach`, and so on), conditions (`ifelse`, `switch`, and so on), functions (`proc`), and built-in Tcl commands (`strings`, `split`, and so on), as well as various Tcl operators.

Additional commands and operators

In addition to standard Tcl syntax, APM supports these commands and operators:

- `mcget` command
- Rule operators: A rule operator compares two operands in an expression.
- Logical operators: A logical operator compares two values in an expression.

Important: *iRules® on the BIG-IP® system can provide functionality to the BIG-IP system components. However, Tcl commands that are specific to iRules are not available in access policy rules.*

Command or Operator	Type	Description
<code>mcget</code>	Command	<code>mcget</code> is an abbreviation for: get the session variable from the memory cache. Access Policy Manager® (APM®) stores all session variables generated in a session in its memory cache. When evaluating a branch rule, APM accesses session variables from system memory using the Tcl command <code>mcget</code> .
<code>contains</code>	Rule operator	Tests whether one string contains another string.
<code>ends_with</code>	Rule operator	Tests whether one string ends with another string.
<code>equals</code>	Rule operator	Tests whether one string equals another string.
<code>matches</code>	Rule operator	Tests whether one string matches another string.
<code>matches_regex</code>	Rule operator	Tests whether one string matches a regular expression.
<code>starts_with</code>	Rule operator	Tests whether one string starts_with another string.
<code>switch</code>	Rule operator	Evaluates one of several scripts, depending on a given value.

Command or Operator	Type	Description
and	Logical operator	Performs a logical and comparison between two values.
not	Logical operator	Performs a logical not action on a value.
or	Logical operator	Performs a logical or comparison between two values.

Tcl examples

These tables describe the syntax elements for the Tcl examples.

Using mcget

```
[ mcget {session.ssl.cert.cn} ]
```

Syntax element	Value	Description
Brackets	[]	The brackets [] that enclose the entire command are the Tcl notation for command evaluation.
Command name	mcget	This command gets the session variable from the memory cache.
Braces	{ }	Braces enclose the session variable.
Session variable name	session.ssl.cert.cn	Session variables that are generated during a session are stored in memory cache.

Checking a certificate field

```
expr { [mcget {session.ssl.cert.OU} ] contains "PD" }
```

This expression checks whether the Organizational Unit (OU) field of a user certificate contains the text PD.

Syntax element	Value	Description
Command name	expr	The Tcl language specifies that an expression begin with the syntax <code>expr</code> .
Rule operator	contains	This operator checks for the string PD.
Return values	0 or 1	0 usually indicates failure, while 1 usually indicates success.

Index

A

access

- date-based 61
- time-based 61

access license usage

- checking from an access policy 63

access policy

- Active Directory authentication 68
- Active Directory query 69
- adding a blank item 23
- adding a HTTP 407 response 32
- adding a logon page 32
- adding a mapping item 27
- adding an assignment item 26
- adding an external logon page 30
- adding an HTTP 401 response 31
- adding an item with configurable properties 25
- adding a preconfigured item 23
- adding a VMware View disclaimer logon page 34
- adding a VMware View SecurID logon page 34
- adding a VMware View Windows logon page 34
- adding connectivity 36, 39
- assigning a local traffic pool 38
- assigning an ACL 36, 39
- assigning an SWG scheme 40–41
- assigning a variable 40
- assigning a virtual keyboard 34
- assigning a webtop 36, 39
- assigning a webtop link 36, 39
- assigning iRule events 66
- certificate inspection 70
- checking NTLM authentication result 72
- configuring date-based access 61
- configuring geolocation-based access 61
- configuring license-based access 63
- configuring time-based access 61
- CRLDP auth action 70
- defining an action 22
- defining an item 22
- defining SSO credential mapping 40
- determining mobile device status 62
- HTTP auth action 70, 73
- IP reputation action 62
- item, adding 15
- Kerberos authentication 71
- LDAP query 71
- localizing a logon page 32
- matching IP subnet 62
- OAM authentication 72
- profile type, related 15
- RADIUS Acct action 74–75
- RADIUS auth action 74
- requesting proxy authentication 32
- RSA SecurID action 75
- SAML authentication 74
- sending email 65
- TACACS+ accounting action 75

access policy (*continued*)

- Transparent Identity Import action 76

access policy branching

- by ActiveSync protocol 57
- by client capable of client-side checks 60
- by client operating system 58
- by client type 59
- by landing URI 62
- by Microsoft Exchange protocol 57

access policy item

- defining 22

access policy result

- session variables 79

access profile type

- action items, related 15

ACL

- assigning from an access policy 36
- dynamic, about 37
- formats supported 37

ACL assign action

- about 36, 39

Active Directory authentication

- session variables 79

Active Directory query

- session variables 79

ActiveSync protocol

- detecting on client 57

ActiveX controls

- and client-side actions 42

AD group resource assign

- in an access policy 36

Advanced Resource Assign

- session variables 79

advanced resource assign action

- about 36, 39

agent-specific information

- in session variable name 78

Android

- operating system check 58

anti-spyware software check

- in an access policy 42

antivirus software check

- dependency on EPSEC software 43
- in an access policy 43

assigning resources

- to active directory groups 36
- to LDAP groups 38

assignment access policy item

- adding dynamically 26
- adding to an access policy 26

assignment items

- resource assignment 35
- variable assignment 35

authentication items

- AAA servers 68
- certificate revocation status 68
- NTLM Auth 68
- RADIUS accounting 68

authentication items *(continued)*
SSL certificates 68
TACACS+ accounting 68

B

BIG-IP Edge Client
client type, detecting 59
BIG-IP Edge Portal
client type, detecting 59
blank access policy item
adding to an access policy 23
blank item
adding dynamically 23
branch rule
advanced example 87
Tcl syntax 86
browser plug-ins
and client-side actions 42
BWC Policy
in an access policy 37

C

Citrix Receiver
client type, detecting 59
Citrix Receiver (legacy)
client type, detecting 59
client certificate authentication
session variables 79
client certificate inspection
client SSL profile, settings for 70
Client-Side Capability action
about 60
client-side checks 60
recommendation 60
Client Type action
Client OS action, compared with 59
supporting multiple traffic types 59
concurrent users
checking from an access policy 63
configurable access policy item
adding dynamically 25
adding to an access policy 25
connectivity license usage
checking from an access policy 63
custom expression
Tcl syntax 86

D

decision box
in an access policy 64
showing at logon 64
Decision box
session variables 79
documentation, finding 18

E

email
adding to an access policy 65
endpoint security
server-side 57
endpoint security (client-side)
benefits 42
purpose 42
endpoint security (client-side) requirements
ActiveX controls 42
browser plug-ins 42
client component installation 42
EPSEC software
and supported antivirus software 43
and supported firewall software 44
and supported hard disk encryption software 45
and supported Health Agent software 53
and supported patch management software 50
and supported peer-to-peer software 51
External Logon page action
about 30

F

File check
session variables 79
files
checking client for existence of 45–46, 53
checking properties of 45–46, 53
on client systems 45–46, 53
firewall software check
dependency on EPSEC software 44
in an access policy 44

G

general purpose items
customizing a log 63
displaying a message 63
presenting two choices 63
processing an iRule 63
reading from a local database 63
sending email 63
writing to a local database 63
guides, finding 18

H

hard disk encryption client-side check
dependency on EPSEC software 45
in an access policy 45
HTTP 401 response
about 31
HTTP 407 response action
about 32

I

in an access policy 53

- iOS
 - operating system check 58
- IP address
 - matching configuring 61
 - matching to physical location 61
- IP address intelligence categories
 - in access policy branch rules 62
- IP reputation
 - in an access policy 62
- iRule events
 - adding to an access policy 66

L

- LDAP authentication
 - session variables 79
- LDAP group resource assign
 - in an access policy 38
- LDAP query
 - session variables 79
- license-based access
 - configuring 63
- Linux
 - operating system check 58
- Linux file check
 - in an access policy 45
- Linux process action
 - in an access policy 46
- local database authentication
 - in an access policy 72
- local traffic pool
 - about assigning 38
- local user database, reading
 - in an access policy 66
- local user database, writing
 - in an access policy 66
- location-based access 61
- logging
 - in an access policy 67
- logon items 30
- Logon Page (CAPTCHA challenge)
 - session variables 79
- logon page action
 - about 32
- logon text
 - protecting with a virtual keyboard 34

M

- Mac
 - operating system check 58
- Mac file check
 - in an access policy 46
- Machine Cert Auth
 - session variables 79
- machine cert auth check
 - in an access policy 47
- Machine info action
 - in an access policy 49
 - Linux support 49
 - MAC address 49
 - Mac support 49

- Machine info action (*continued*)
 - Windows support 49
- Mac process action
 - in an access policy 47
- macro
 - about 14, 17
 - adding as an access policy action 16
 - terminal, about 14, 17
- macrocall
 - about 14, 17
 - adding to access policy 14, 17
- macrocalls
 - on add item screen 16
- manuals, finding 18
- mapping access policy item
 - adding dynamically 27
 - adding to an access policy 27
- mcget command 86
- message box
 - in an access policy 67
 - showing at logon 67
- Microsoft Exchange protocol
 - detecting on client 57
- mobile browser
 - client type, detecting 59
- mobile device
 - determining jailbroken status 62
 - determining rooted status 62

N

- NTLM authentication result
 - checking, from an access policy 72

O

- on-demand certificate authentication
 - about 73
 - configuring for iPhone 73
 - configuring for iPod 73
 - Safari, behavior with 73
- operating system check 58
- OTP Generate
 - about 73
 - session variables 79
- OTP Verify
 - about 74
 - session variables 79

P

- patch management software check
 - dependency on EPSEC software 50
 - in an access policy 50
- peer-to-peer software check
 - dependency on EPSEC software 51
 - in an access policy 51
- preconfigured access policy item
 - adding to an access policy 23
- preconfigured item
 - adding dynamically 23

R**RADIUS**

- session variables [79](#)

- release notes, finding [18](#)

- resource allocation

- session variables [79](#)

- resource assign action

- about [39](#)

- route domain

- selecting in an access policy [39](#)

S

- session management

- session variables [79](#)

- session variable

- assigning [40](#)

- session variable name

- \$attr [78](#)

- \$name [78](#)

- last [78](#)

- session variables

- about [78](#)

- in reports [78](#)

- logging in an access policy [67](#)

- using in branch rules [78](#)

- viewing [78](#)

SNAT

- assigning in an access policy [39](#)

- assigning in network access resource [39](#)

- assigning in virtual server [39](#)

- assignment precedence [39](#)

- split domain

- and session.logon.last.username [30](#)

- SSO credentials sAMAccountName

- session variable for [40](#)

- specifying [40](#)

- specifying for SSO [40](#)

T

- Tcl

- about usage [86](#)

- Tcl expressions

- and Variable Assign item [35](#)

- Tcl syntax

- and iRules [86](#)

- APM logical operators [86](#)

- APM rule operators [86](#)

- brace [87](#)

- bracket [87](#)

- expr command [87](#)

- mcget command [86](#)

V

- variable

- about assigning [40](#)

- variable assign

- custom expression example [87](#)

- virtual keyboard

- about [34](#)

- visual policy editor branches, about

- endings, configuring [14](#)

- macro, adding and configuring [14](#)

- macrocall, adding [14](#)

- rectangle, about [14](#)

- red asterisk, about [14](#)

- shaded rectangle, about [14](#)

- VMware View

- client type, detecting [59](#)

- VMware View logon page action

- about [34](#)

W

- web browser

- client type, detecting [59](#)

- webtop and links assign action

- about [36](#), [39](#)

- Windows

- operating system check [58](#)

- Windows cache and session control

- and Windows Protected Workspace [52](#)

- cleaning up after an access session [52](#)

- in an access policy [52](#)

- Windows file check [53](#)

- Windows Health Agent action

- dependency on EPSEC software [53](#)

- in an access policy [53](#)

- Windows Inbox F5 VPN Client

- client type, detecting [59](#)

- Windows Info

- session variables [79](#)

- Windows operating system

- service packs, checking [54](#)

- updates, checking [54](#)

- verifying [54](#)

- Windows Process

- session variables [79](#)

- Windows process action

- in an access policy [54](#)

- Windows protected workspace

- and Windows Cache and Session Control [55](#)

- in an access policy [55](#)

- Windows registry

- in an access policy [56](#)

- Windows Registry

- session variables [79](#)

- Windows Registry check

- 32-bit and 64-bit registry keys [57](#)

- in an access policy [57](#)

- supported registry keys [57](#)

- Windows service pack

- checking client [54](#)