

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

Version 13.0



Table of Contents

Visual Policy Editor.....	7
About the visual policy editor.....	7
Visual policy editor conventions.....	7
About actions on the add item screen.....	8
About macrocalls on the add item screen.....	9
About macros and macrocalls.....	10
About maximum depth for nested macros in an access policy.....	11
About access policy endings.....	11
About maximum expression size for visual policy editor.....	12
About per-session and per-request policies.....	12
About per-request policies and the Apply Access Policy link.....	12
About per-request policies and nested macros.....	12
About per-request policy subroutines.....	12
About subsessions.....	12
Additional resources and documentation for BIG-IP Access Policy Manager.....	13
 Defining Access Policy Items.....	 15
About access policy item configuration.....	15
Adding a blank access policy item to an access policy.....	16
Adding an access policy item with preconfigured branch rules.....	17
Adding an access policy item with configurable properties.....	18
Adding an access policy assignment item.....	20
Adding an access policy mapping item.....	21
 Access Policy Item Reference.....	 23
About logon items.....	23
About the External Logon page.....	23
About HTTP 401 Response	24
About HTTP 407 Response.....	25
About Logon Page.....	25
About the virtual keyboard.....	27
About the VMware View logon page action.....	28
About assignment items.....	29
About ACL Assign	29
About AD Group Resource Assign	29
About Advanced Resource Assign	30
About BWC Policy	30
About Citrix Smart Access	30
About Dynamic ACL	31
About LDAP Group Resource Assign	31
About Pool Assign	31
About Resource Assign	32
About Route Domain and SNAT Selection	32
About SSO Credential Mapping	33
About Variable Assign	33
About VMware View Policy.....	34
About Webtop, Links and Sections Assign	35
About endpoint security client-side items.....	35

About client-side action requirements and alternatives.....	35
About the Anti-spyware action.....	35
About the Antivirus action.....	36
About the Firewall action.....	37
About Hard Disk Encryption	38
About Linux File	39
About Linux Process	39
About Mac File	40
About Mac Process	40
About Machine Cert Auth	41
About Machine Info	43
About Patch Management	44
About Peer-to-Peer	45
About Windows Cache and Session Control	46
About the Windows File action.....	46
About Windows Health Agent	47
About Windows Info	48
About Windows Process	48
About Windows Protected Workspace	49
About Windows Registry	50
About 32-bit registry keys on a 64-bit Windows client.....	51
About endpoint security (server-side) access policy items.....	52
About Client for MS Exchange	52
About Client OS	52
About Client Type	53
About Client-Side Capability	55
About the Date Time action.....	55
About IP Geolocation Match	56
About IP Reputation	56
About IP Subnet Match	56
About Jailbroken or Rooted Device Detection	57
About Landing URI	57
About the License action.....	57
About general purpose items.....	58
About the Decision Box action.....	58
About the Email action.....	60
About the Empty action.....	61
About iRule Event	61
About Local Database	61
About the Logging action.....	62
About the Message Box action.....	63
About authentication items.....	63
About AD Auth	64
About AD Query	65
About Client Cert Inspection	66
About CRLDP Auth	66
About HTTP Auth	66
About Kerberos Auth	66
About LDAP Query	67
About LocalDB Auth	67
About NTLM Auth Result	67
About the OAuth Authorization.....	68
About OAuth Client.....	68
About OAuth Scope.....	69
About OCSP Auth	69
About On-Demand Cert Auth	70

About OTP Generate	70
About OTP Verify	70
About SAML Auth	71
About RADIUS Acct	71
About RADIUS Auth	71
About RSA SecurID	72
About TACACS+ Acct	72
About TACACS+ Auth.....	72
About Transparent Identity Import.....	73
Per-Request Policy Item Reference.....	75
About per-request policy items.....	75
About Protocol Lookup.....	75
About SSL Bypass Set.....	75
About AD Group Lookup.....	75
About LDAP Group Lookup.....	75
About LocalDB Group Lookup.....	75
About RADIUS Class Lookup.....	76
About Dynamic Date Time.....	76
About SSL Intercept Set.....	77
About the Logging action.....	77
About Category Lookup.....	77
About Response Analytics.....	78
About Request Analytics.....	79
About URL Filter Assign.....	79
About Application Lookup	79
About Application Filter Assign.....	80
About HTTP Headers.....	80
About Select SSO Configuration.....	81
About OAuth Client.....	81
About OAuth Scope.....	82
About per-request policy subroutine items.....	82
Per-session policy and subroutine agent differences.....	82
About AD Auth	83
About Confirm Box.....	84
About CRLDP Auth	84
About HTTP Auth	85
About HTTP 401 Response	85
About iRule Event	86
About LDAP Auth	86
About LocalDB Auth	87
About Logon Page.....	87
About On-Demand Cert Auth	89
About OAuth Client.....	89
About OAuth Scope.....	90
About OCSP Auth	91
About Proxy Select.....	91
About RADIUS Auth	91
About Select SSO Configuration.....	92
About URL Branching.....	92
About per-request policy endings.....	93
Session Variables.....	95
About session variables.....	95
About session variable names.....	95

Session variables reference.....	96
sessiondump command usage.....	101
Perflow Variables.....	103
Per-flow and subsession variables.....	103
Per-Request Policy Maintenance.....	107
Customizing messages for the per-request policy Reject ending.....	107
Exporting and importing a per-request policy across BIG-IP systems.....	107
Tcl Usage.....	109
About Tcl usage in APM.....	109
Tcl syntax notes.....	109
Tcl examples.....	110
Variable Assign Reference.....	111
Variable assign: domain plus username example.....	111
Variable assign: text assignment example.....	111
Predefined session variable attributes.....	112
Windows Registry Reference.....	115
Overview: Policy branching based on Windows Registry values.....	115
Registry screenshot: Allowed keys for a trusted server.....	115
Example: Allowed registry key value fetched.....	115
Configuring clients for Windows Registry GET operation.....	116
Viewing trusted server registry keys and subkeys on a client.....	117
Fetching the value of a Windows Registry key from a client.....	117
Legal Notices.....	119
Legal notices.....	119

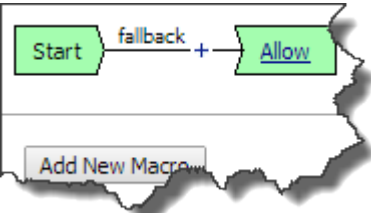

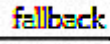

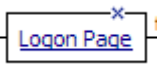

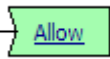
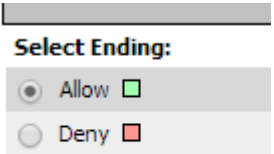
Visual Policy Editor

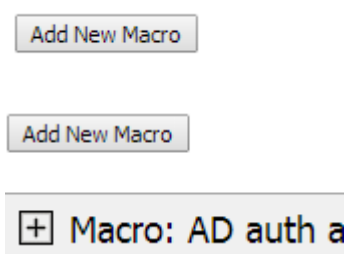
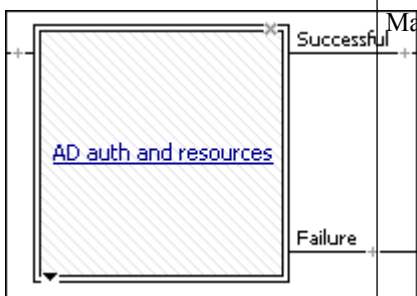
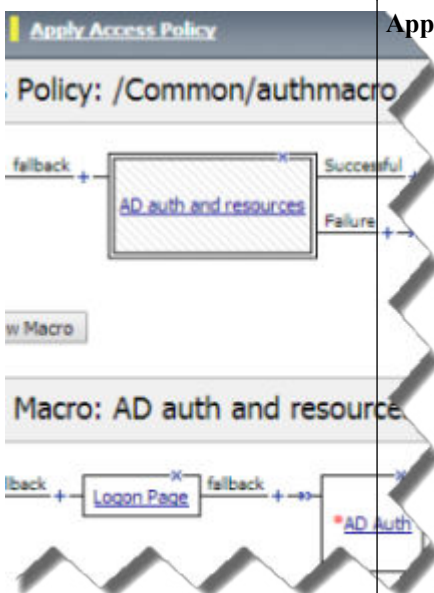
About the visual policy editor

The visual policy editor is a screen on which to configure a per-session policy (also known as an access policy) or a per-request 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 and initial per-request policy	When an access profile is created, usually an initial access policy is also created. A per-request policy starts with similar initial elements.
	Start	Every access policy and per-request policy 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. An access policy includes Allow or Deny endings. A per-request policy includes Allow or Reject endings.
	Configure ending	Clicking the name of an ending opens a popup screen.

Visual element	Element type	Description
	<p>Add a macro for use in the access policy</p> <p>Macro added for use</p>	<p>Opens a screen for macro template selection. After addition, the macro is available for configuration and for use as an action item.</p> <p>Added macros display under the access policy. Clicking the plus (+) sign expands the macro for configuration of the actions in it.</p>
	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.

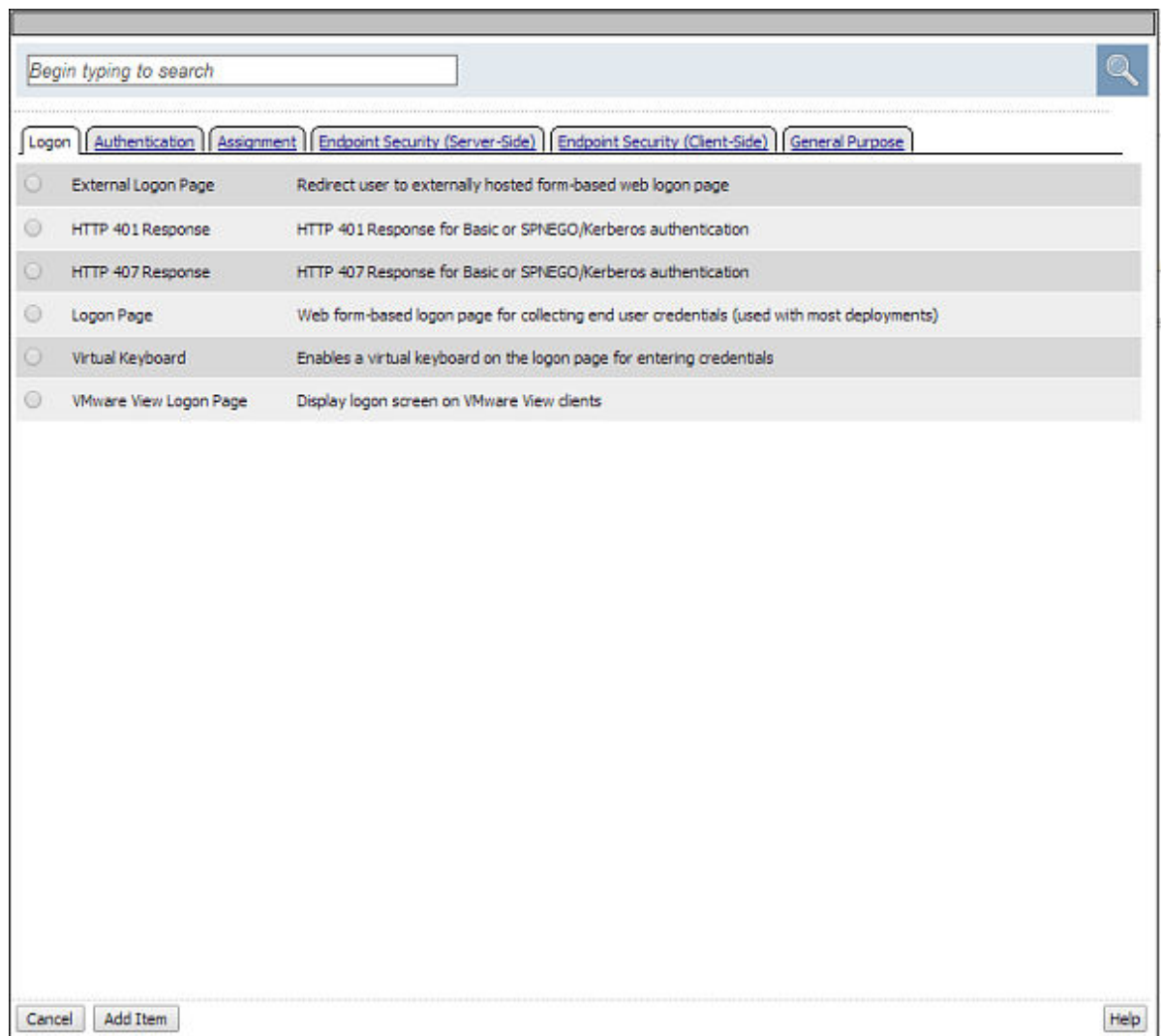


Figure 1: Add action item screen

About macrocalls on the add item screen

The Macrocalls tab displays when one or more macros has been added for use in the access policy. When adding an access policy item to a macro, the Macrocalls tab displays unless adding a macrocall would create a misconfiguration, such as causing a macro loop or causing a series of macrocalls to exceed a depth of three.

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

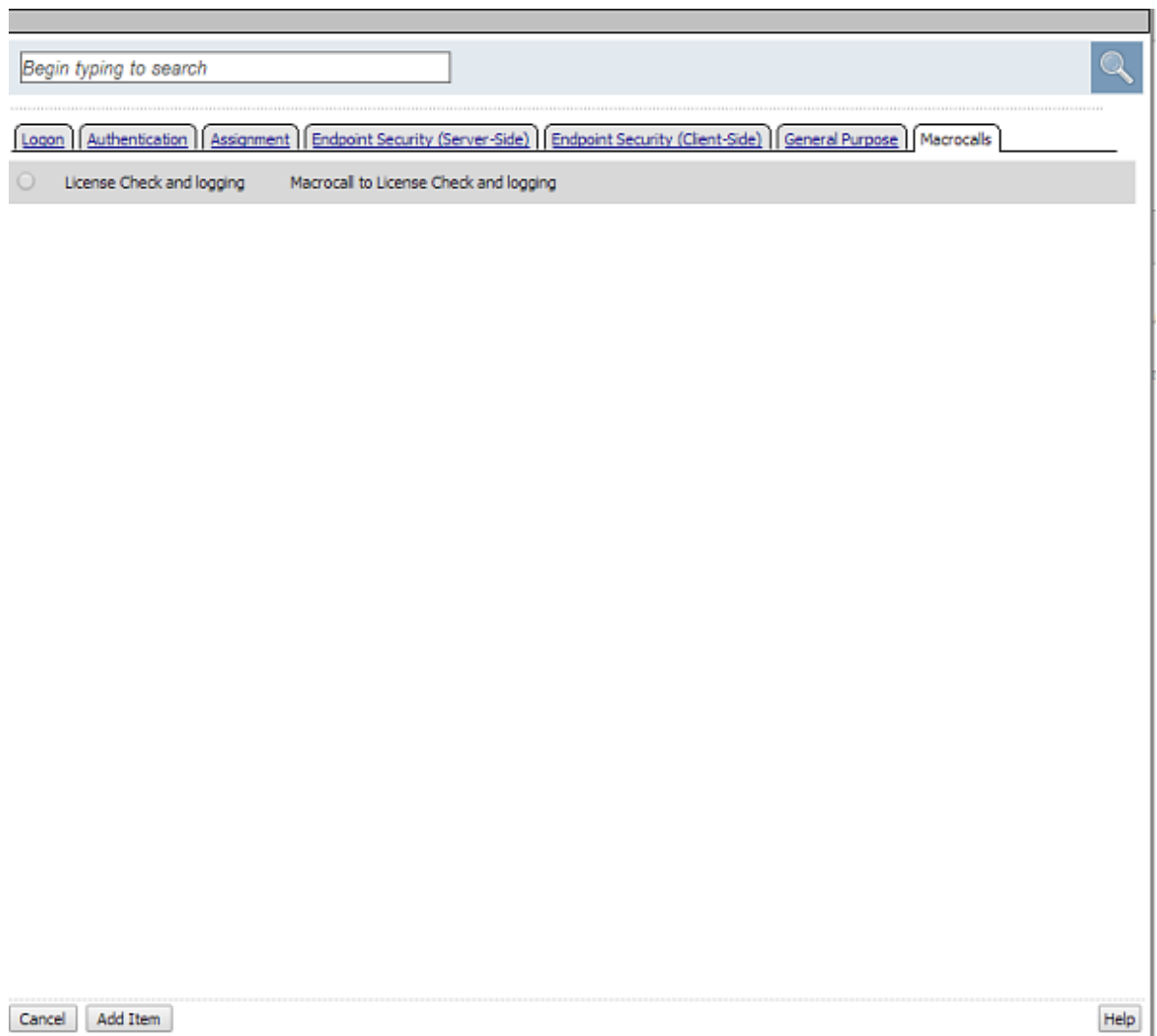


Figure 2: Macrocalls tab on the add item screen

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.

About maximum depth for nested macros in an access policy

In an access policy, a macro can make a macrocall to another macro until up to three macros have been called in series.

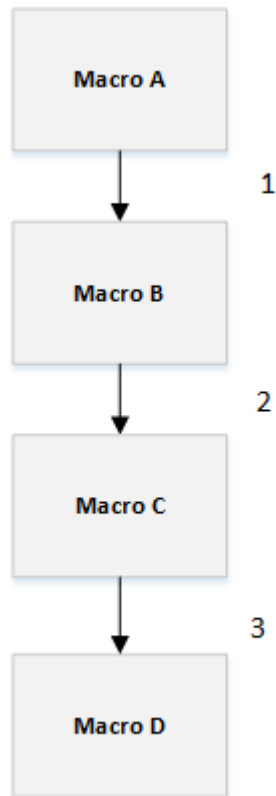


Figure 3: The maximum depth of macrocalls

About access policy endings

An ending provides a result for an access policy branch. An ending for an access policy branch is one of three types.

Allow

Starts the SSL VPN session and loads assigned resources and a webtop, if assigned, for the user. Typically, you assign this when the user passes specific checks.

Deny

Disallows the SSL VPN session and shows the user an access denied web page. Typically, you assign this when the user does not have access to resources, or fails authentication. Alternatively after a session starts, shows a URL filter denied web page after a per-request policy rejects a request for a URL.

Redirect

Redirects the client to the URL specified in the ending configuration. You can define a redirect URL for each redirect ending. Typically, you can assign a redirect when the user requires remediation, or a separate resource. For example, a user who fails the antivirus check because virus definitions are out of date can be redirected to the software manufacturer's site to get an antivirus update.

About maximum expression size for visual policy editor

The maximum size for an expression in the visual policy editor is 64 KB. The visual policy editor cannot save an expression that exceeds this limit.

About per-session and per-request policies

Access Policy Manager® (APM®) provides two types of policies.

Per-session policy

The per-session policy runs when a client initiates a session. (A per-session policy is also known as an access policy.) Depending on the actions you include in the access policy, it can authenticate the user and perform other actions that populate session variables with data for use throughout the session.

Per-request policy

After a session starts, a *per-request policy* runs each time the client makes an HTTP or HTTPS request. A per-request policy can include a subroutine, which starts a subsession. Multiple subsessions can exist at one time.

One access policy and one per-request policy are specified in a virtual server.

About per-request policies and the Apply Access Policy link

The Apply Access Policy link has no effect on a per-request policy. Conversely, updates made to a per-request policy do not affect the state of the Apply Access Policy link.

About per-request policies and nested macros

Access Policy Manager® (APM®) supports calling a macro from a per-request policy and calling a subroutine macro from a per-request policy subroutine. However, APM does not support calling any type of macro from a per-request policy macro or from a per-request policy subroutine macro.

About per-request policy subroutines

A per-request policy *subroutine* is a collection of actions. What distinguishes a subroutine from other collections of actions (such as macros), is that a subroutine starts a subsession that, for its duration, controls user access to specified resources. Subroutine properties not only specify resources but also specify subsession timeout values and maximum subsession duration.

About subsessions

A *subsession* starts when a subroutine runs and continues until reaching the maximum lifetime specified in the subroutine properties, or until the session terminates. A subsession does not count against license limits. A subsession populates subsession variables that are available for the duration of the subsession.

Subsession variables and events that occur during a subsession are logged. Multiple subsessions can exist at the same time.

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®: 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.
<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®: 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.
<i>BIG-IP® Access Policy Manager®: Edge Client and Application Configuration</i>	This guide contains information for an administrator to configure the BIG-IP® system for browser-based access with the web client as well as for access using BIG-IP Edge Client® and BIG-IP Edge Apps. It also includes information about how to configure or obtain client packages and install them for BIG-IP Edge Client for Windows, Mac, and Linux, and Edge Client command-line interface for Linux.
<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®: 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®: 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®: Secure Web Gateway</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</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.

Document	Description
<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.
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.

Defining Access Policy Items

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 verify a wide range of conditions with Expression screens.	<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 verify 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 actions, such as Webtop, Links and Sections 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, Links and Sections 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 > Profiles / Policies**.
The Access Profiles (Per-Session Policies) screen opens.
2. In the Per-Session Policy column, click the **Edit** link for the access profile you want to configure.
The visual policy editor opens the access policy in a separate screen.
3. Click the (+) icon anywhere in the access policy to add a new item.

***Note:** Only an applicable subset of access policy items is available for selection in the visual policy editor for any access profile type.*

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 area.
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 policy 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.

***Note:** To ensure that logging is configured to meet your requirements, verify the log settings for the access profile.*

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 > Profiles / Policies**.
The Access Profiles (Per-Session Policies) screen opens.
2. In the Per-Session Policy column, click the **Edit** link for the access profile you want to configure.
The visual policy editor opens the access policy in a separate screen.
3. Click the (+) icon anywhere in the access policy to add a new item.

***Note:** Only an applicable subset of access policy items is available for selection in the visual policy editor for any access profile type.*

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 policy 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.

Note: To ensure that logging is configured to meet your requirements, verify the log settings for the access profile.

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 > Profiles / Policies**.
The Access Profiles (Per-Session Policies) screen opens.
2. In the Per-Session Policy column, click the **Edit** link for the access profile you want to configure.
The visual policy editor opens the access policy in a separate screen.
3. Click the (+) icon anywhere in the access policy to add a new item.

Note: Only an applicable subset of access policy items is available for selection in the visual policy editor for any access profile type.

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.

Option	Description
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.
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 policy 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.

***Note:** To ensure that logging is configured to meet your requirements, verify the log settings for the access profile.*

Adding an access policy assignment item

Before you can add an access policy assignment item, you need to 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.

1. On the Main tab, click **Access > Profiles / Policies**.
The Access Profiles (Per-Session Policies) screen opens.
2. In the Per-Session Policy column, click the **Edit** link for the access profile you want to configure.
The visual policy editor opens the access policy in a separate screen.
3. Click the (+) icon anywhere in the access policy to add a new item.

***Note:** Only an applicable subset of access policy items is available for selection in the visual policy editor for any access profile type.*

A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.

4. 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 > Webtop, Links and Sections Assign	Allows you to assign a webtop, webtop links, and webtop sections to an access policy branch.

A properties screen opens.

5. Configure the properties for the item.
6. Click **Save**.
The properties screen closes and the policy 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.

Note: To ensure that logging is configured to meet your requirements, verify the log settings for the access profile.

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.

1. On the Main tab, click **Access > Profiles / Policies**.
The Access Profiles (Per-Session Policies) screen opens.
2. In the Per-Session Policy column, click the **Edit** link for the access profile you want to configure.
The visual policy editor opens the access policy in a separate screen.
3. Click the (+) icon anywhere in the access policy to add a new item.

Note: Only an applicable subset of access policy items is available for selection in the visual policy editor for any access profile type.

A popup screen opens, listing predefined actions on tabs such as General Purpose, Authentication, and so on.

4. 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 policy 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.

Note: To ensure that logging is configured to meet your requirements, verify the log settings for the access profile.

Access Policy Item Reference

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.

***Note:** For a per-request policy subroutine, HTTP 401 Response supports HTTP Basic authentication only.*

The HTTP 401 Response action provides up to 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.

Basic Auth Realm

Specifies the authentication realm for use with Basic authentication.

HTTP Auth Level

Specifies the authentication required for the policy.

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

***Note:** This option is not available for a per-request policy subroutine.*

- **basic+negotiate** - specifies either Basic or Kerberos authentication.

***Note:** This option is not available for a per-request policy subroutine.*

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

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.

***Note:** When configured in a per-request subroutine, some screen elements and options described here might not be available.*

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 check box.
- **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 (or Subsession 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`.

***Note:** A per-request policy subroutine uses subsession variables in place of session variables.*

Clean Variable

Specifies whether to clear any value from the variable before presenting the logon page to the user; to clean the variable, select **Yes**. Defaults to **No**.

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 to 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, Webtop Sections, 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 agent can dynamically assign a local traffic pool; it provides this configuration element only: selection of a static pool.

In a per-session policy, the Pool Assign agent enables session-based pool selection 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 agent takes precedence over a static pool defined for the virtual server.

In a per-request policy, the Pool Assign agent enables request-based pool selection for reverse proxy (LTM+APM) only. In a per-request policy, the pool specified in Pool Assign agent is assigned.

***Note:** In a per-request policy, using the Pool Assign agent in a forward proxy configuration does not work and is not supported.*

About Resource Assign

The Resource Assign action assigns connection resources to a session.

A Resource Assign action provides these configuration elements and options:

Network Access

Specifies the names of one or more network access resources.

Portal Access

Specifies the names of one or more portal access resources.

App Tunnel

Specifies the names of one or more application tunnels.

Remote Desktop

Specifies the names of one or more remote desktops.

SAML

Specifies the names of one or more SAML resources.

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 Variable Assign

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

Figure 4: 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 a session variable or the name of a perflow variable.

***Note:** For a per-session policy, when the policy runs it recognizes only existing perflow variables.*

Predefined Variables

Specifies a predefined session variable or perflow variable name, which must be selected from the **Variable** list. The type of variable (session or perflow) that is available for selection depends on the selected **Group**: **Per-Session Variables** or **Per-Request Variables**.

For **Per-Request Variables**, the **Scratchpad**, **Custom**, and **Primary Category** perflow variables are available for use in the per-request policy and in per-request policy subroutines. You can, for example, pass the value of a session variable into the per-request policy in one of these variables.

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.

A 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 depending on the agent type:
 - **Use user's attribute** - for AD agent.
 - **Use user's primary group attribute** - for AD agent.
 - **Use LDAP attribute** - for LDAP agent.
 - **Use RADIUS attribute** - for RADIUS agent.
- **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 VMware View Policy

A VMware View Policy action provides the ability to enable USB redirection for View desktop resources, and to pass Start Session Script Variables for VMware View connections for supported View clients.

A VMware View Policy action provides these configuration elements and options:

USB redirection

- **Disabled** - Disables USB redirection. This is the default setting.
- **Enabled** - Enables USB redirection.

VMware View Start Session Script Variables

Specify variables and values to pass to the VMware View Connection Server (VCS) for use in a Start Session Script that you have configured on the VCS.

About Webtop, Links and Sections Assign

The Webtop, Links and Sections Assign action can assign a webtop, preconfigured webtop links, and webtop sections to a session.

A Webtop, Links and Sections Assign 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.

Webtop Sections

Specifies one or more webtop sections for grouping links on the webtop.

Note: Webtop sections apply only to a full 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**.

***Note:** When **Enabled**, if the client does not respond for five minutes, the server ends the session.*

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**.

***Note:** When **Enabled**, if the client does not respond for five minutes, the server ends the session.*

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**.

***Note:** When **Enabled**, if the client does not respond for five minutes, the server ends the session.*

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**.

***Note:** When **Enabled**, if the client does not respond for five minutes, the server ends the session.*

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**.

***Note:** When **Enabled**, if the client does not respond for five minutes, the server ends the session.*

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**.

***Note:** When **Enabled**, if the client does not respond for five minutes, the server ends the session.*

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**.

***Note:** When **Enabled**, if the client does not respond for five minutes, the server ends the session.*

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**.

***Note:** When **Enabled**, if the client does not respond for five minutes, the server ends the session.*

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.

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**, a UAC prompt for users with admin-level privileges is allowed. When set to **No** the UAC prompt for non-admin users is suppressed, which can cause a failure to verify the machine certificate. This setting does not affect users without admin-level privileges. If the Machine Certificate Checker Service is installed and **Allow User Account Control right elevation prompts** is set to **Yes**, the following scenarios occur:

- Users with administrator privilege are prompted for UAC.
- Standard users who use Machine Certificate Checker service will not be prompted for UAC.
- Guest users who use Machine Certificate Checker service will not be prompted for UAC.

If the Machine Certificate Checker Service is installed and **Allow User Account Control right elevation prompts** is set to **No**, the following scenarios occur:

- Users with administrator privilege are not prompted for UAC.
- Standard users who use Machine Certificate Checker service will not be prompted for UAC.
- Guest users who use Machine Certificate Checker service will not be prompted for UAC.

If you do not install Machine Certificate Checker Service and set **Allow User Account Control right elevation prompts** to **Yes**, the following scenarios occur:

- Users with administrator privilege are prompted for UAC.
- Standard users will fail to verify machine certificate.
- Guest users will fail to verify machine certificate.

If you do not install Machine Certificate Checker Service and set **Allow User Account Control right elevation prompts** to **No**, the following scenarios occur:

- Users with administrator privilege will fail to verify machine certificate.
- Standard users will fail to verify machine certificate.
- Guest users will fail to verify machine certificate.

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	<code>session.machine_info.cpu.name</code>
CPU Vendor ID	<code>session.machine_info.cpu.vendor</code>
CPU Description	<code>session.machine_info.cpu.description</code>
CPU maximum clock	<code>session.machine_info.cpu.max_clock</code>
Motherboard manufacturer	<code>session.machine_info.motherboard.manufacturer</code>
Motherboard serial number	<code>session.machine_info.motherboard.sn</code>
Motherboard product	<code>session.machine_info.motherboard.product</code>
BIOS manufacturer	<code>session.machine_info.bios.manufacturer</code>

Information	Session variable name
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**.

***Note:** When **Enabled**, if the client does not respond for five minutes, the server ends the session.*

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**.

***Note:** When **Enabled**, if the client does not respond for five minutes, the server ends the session.*

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**.

***Note:** When **Enabled**, if the client does not respond for five minutes, the server ends the session.*

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**.

***Note:** When **Enabled**, if the client does not respond for five minutes, the server ends the session.*

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**.

***Note:** When **Enabled**, if the client does not respond for five minutes, the server ends the session.*

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 nameGroupA. However, if the user then connects to a server with persistent storage

enabled, and the server group name GroupB, persistent data from the GroupAProtected 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. Windows Registry can also fetch the value of a key and store it in a session variable, provided that the client is configured to allow the value to be fetched.

The Windows Registry action provides these configuration elements:

Continuously check the result and end the session if it changes

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

***Note:** When **Enabled**, if the client does not respond for five minutes, the server ends the session.*

Expression

Specifies a Boolean expression.

This is the syntax for registry checker expressions:

```
"key" comparison_operator data
"key"."value" >> "variable_name"
"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 (=) and the double equal sign (==) specify equality.

***Note:** Windows Registry does not support comparison of binary data.*

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*.

>>

This is the GET operator; it fetches the value of a key provided that the client is configured to allow the value to be fetched.

***Note:** The GET operator supports these Windows Registry data types only: REG_DWORD, REG_SZ, and REG_MULTI_SZ. The maximum recommended amount of data that a GET operator returns should be kept within 1 KB.*

All sub-expressions of a compound expression are evaluated even if they are joined by OR operators. Windows Registry values are sent to the server for successful fetch operations even if the overall expression is evaluated to false.

"variable_name"

Represents the variable into which the GET operator stores the value of the key. When a GET operation is successful, the registry value retrieved from the client is saved in a session variable in this format: `session.windows_check_registry.last.data.variable_name`. Quotation marks are required. The variable name can include alphanumeric symbols, underscores, dots, and hyphens only and must be no more than 64 characters long.

Table 2: Example expressions

Expression	Description
<code>"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer". "Build">>"IEBuild"</code>	Checks for the presence of the specified path in the registry and stores the value in the variable <code>session.windows_check_registry.last.data.IEBuild</code> .
<code>"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\InternetExplorer". "Version">= "6.0.2900.2180"</code>	Checks that the Internet Explorer version is greater than or equal to the value specified.
<code>"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\InternetExplorer". "Version">= "5.0.2800.0" AND "HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\InternetExplorer". "Version" >= "6.0.2900.0"</code>	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.

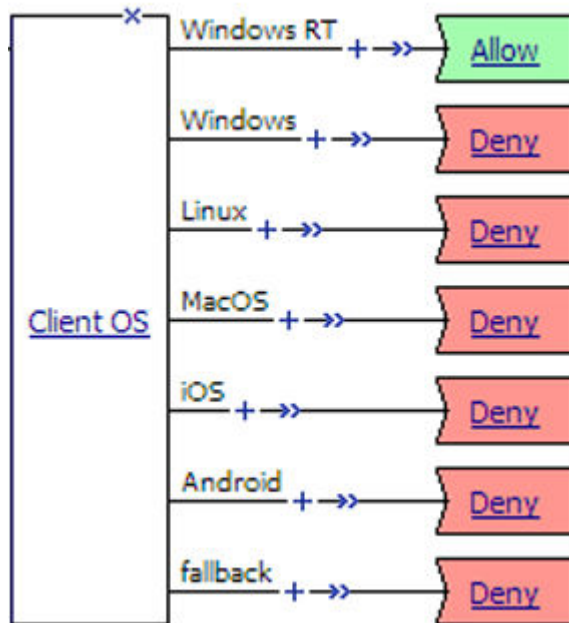


Figure 5: Client OS item with Allow ending configured on Windows RT branch

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.

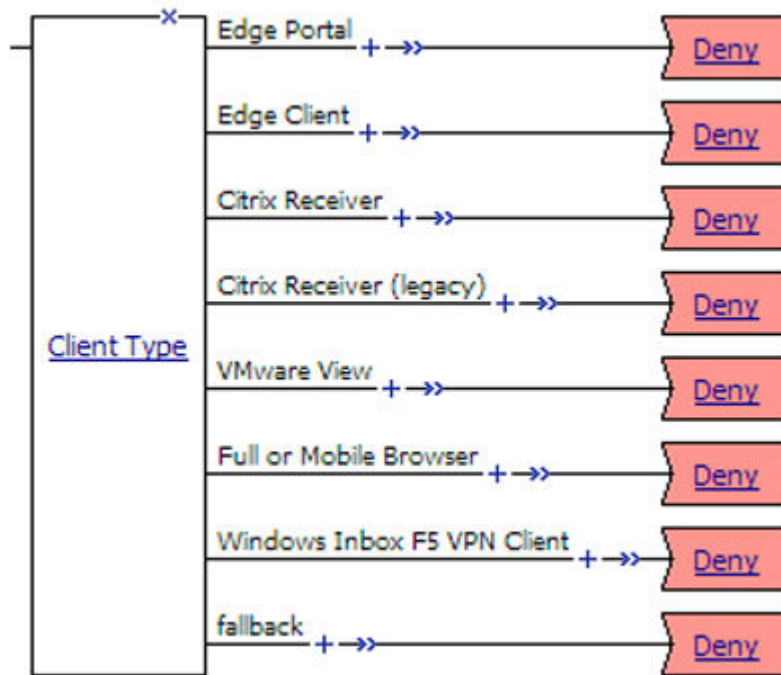


Figure 6: Client Type

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.

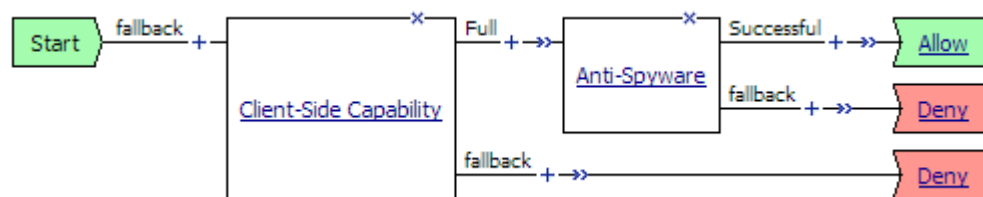


Figure 7: Client-side capability check before a client-side check

***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.

***Note:** The Jailbroken or Rooted Device Detection action works only while using F5 Access or BIG-IP® Edge Client®.*

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.

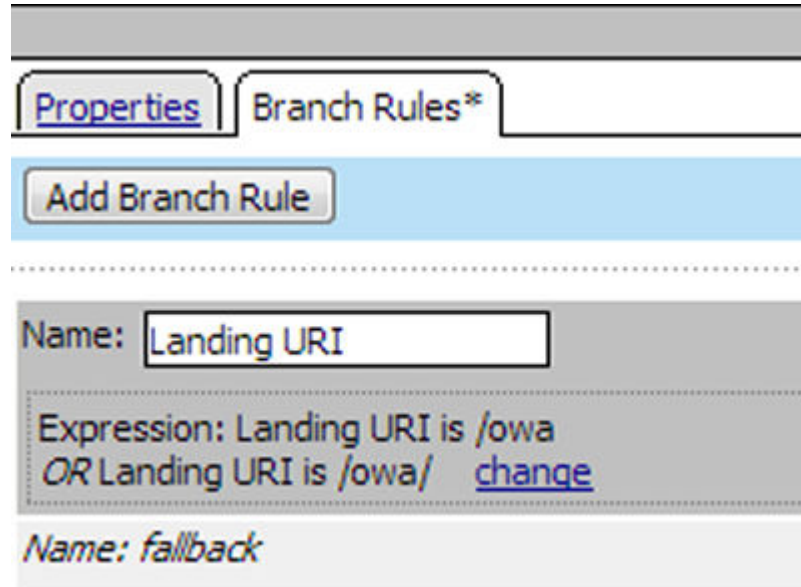


Figure 8: Landing URI branch rule with updated expression

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 the 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 access licenses for the access profile against the number that you specify.
- **Remaining Concurrent User count is above percentage threshold** - checks percentage of concurrent access licenses for the access profile 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 Visual Policy Editor interface is shown with a policy flow at the top and a configuration panel below.

Policy Flow:

- The flow starts with an **Anti-Spyware** check.
- If **Successful**, it proceeds to an **Allow** ending.
- If **fallback**, it proceeds to a **Decision Box**.
- From the **Decision Box**:
 - If **Guest net**, it proceeds to **Advanced Resource Assign**.
 - If **fallback**, it proceeds to a **Deny** ending.
- From **Advanced Resource Assign**:
 - If **fallback**, it proceeds to an **Allow** ending.

Configuration Panel:

Properties* | Branch Rules

Name:

Decision Box

Customization

Language	en ▼	Reset all defaults
Message	Your anti-spyware software is not up-to-date. Please choose one of these options:	
Field 1 image	green icon ▼	
Option 1	Log on to the Guest network.	
Field 2 image	red icon ▼	
Option 2	Log out	

Figure 9: 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.

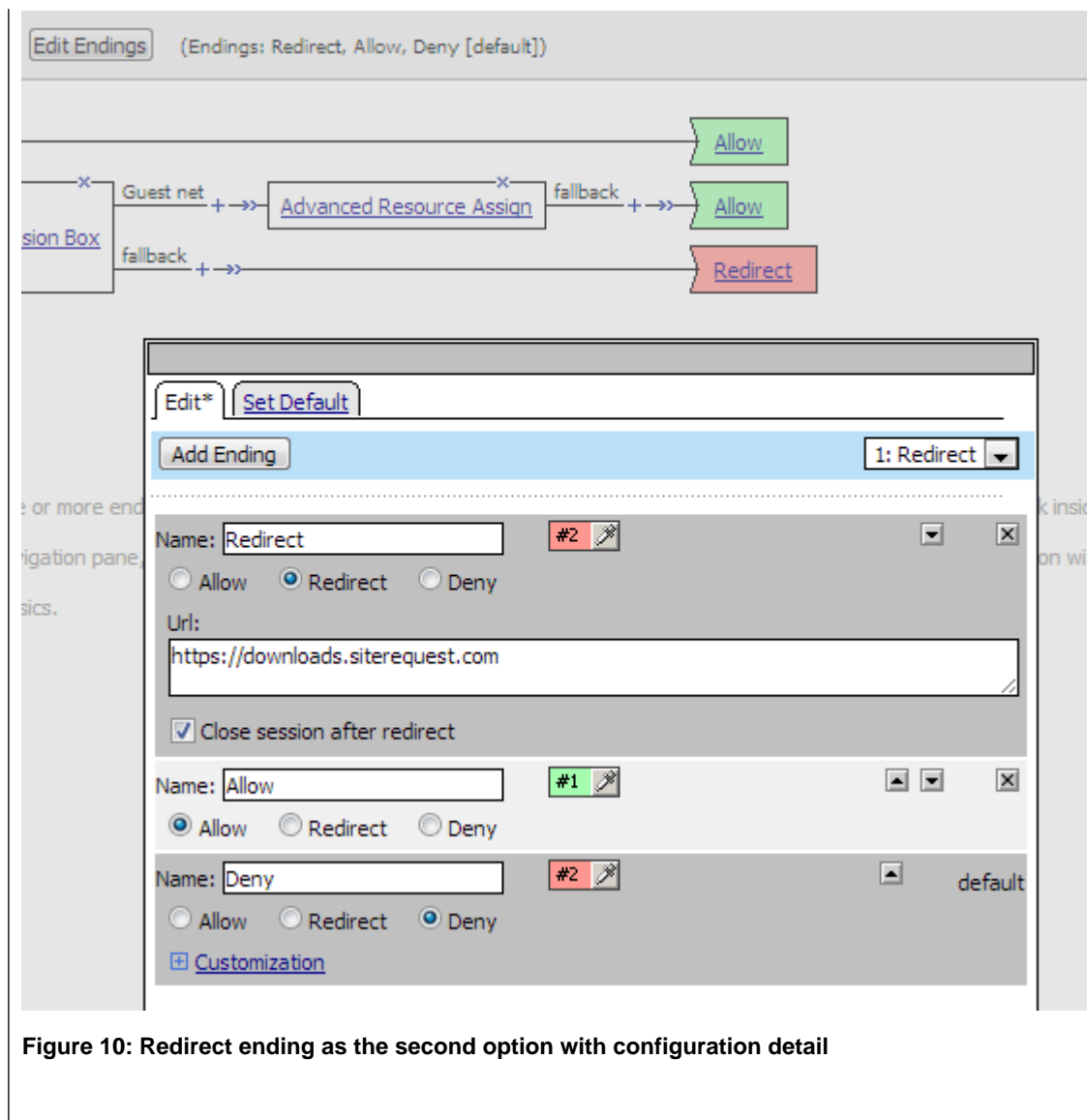


Figure 10: Redirect ending as the second option with configuration detail

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 or to a per-request policy subroutine 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 or a per-request policy subroutine.

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, **Destination** specifies a variable; the value that is read from the database is stored in this variable. The default variables are:

- `session.localdb.groups`
- `session.localdb.locked_out`
- `session.localdb.login_failures`

***Note:** Alternatively, the variable can be any text string. When using non-default variables, verify the expressions used in any other item that reads or manipulates local database variables in the same session; ensure that the expressions use the same string.*

For the **Write** action, **Destination** 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) to read. For **Write**, specifies an expression, the value of which will be written.

About the Logging action

The Logging action can be used in an access policy or in a per-request policy. In an access policy, the Logging action adds logging for session variables to the access policy. In a per-request policy, the Logging action can add logging for both session variables and per flow variables to the per-request policy.

This action is useful for tracing the 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

For an access policy, specifies text to add to the log file. For a per-request policy, specifies the message text and the session and per-flow variables to add to the message. Complete variable names must be typed. Wildcards are not supported for per-request policies. An example log message for a per-request policy follows.

```
The system found this URL %{perflow.category_lookup.result.url} in these categories %
{perflow.category_lookup.result.categories} and placed it into this category %
{perflow.category_lookup.result.primarycategory}.
```

```
An HTTPS request was made to this host %{perflow.category_lookup.result.hostname}; the per-
request policy set SSL bypass to %{perflow.ssl_bypass_set}.
```

```
Requests from this platform %{session.client.platform} were made during this session %
{perflow.session.id}.
```

Session Variables

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

***Note:** This option is available only when adding the Logging action to an access policy.*

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. An authentication action typically follows a logon action that collects credentials.

***Note:** When configured in a per-request policy subroutine, some screen elements and options described here might not be available.*

Type

Specifies Authentication, the type of this Active Directory action.

Server

Specifies an Active Directory server; servers are defined in the **Access > Authentication** 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.

***Note:** For a per-request policy subroutine, equivalent functionality is supported through subroutine settings.*

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 > Authentication** 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 Group1 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 > Authentication** area of the Configuration utility.

Note: A CRLDP Auth action is valid for use in a per-request policy subroutine when placed after an On-Demand Cert Auth action.

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 > Authentication** 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 > Authentication** 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.

Note: For a per-request policy subroutine, equivalent functionality is supported through subroutine settings.

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 > Authentication** 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 Group1 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.

Note: For a per-request policy subroutine, equivalent functionality is supported through subroutine settings.

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 the OAuth Authorization

When Access Policy Manager® (APM®) is configured to act as an OAuth authorization server, an OAuth Authorization agent must be present in the access policy.

The OAuth Authorization agent provides these elements and options.

Prompt for Authorization

- **Enabled** - Displays the OAuth Authorization page. The page requests authorization for the client application to access a list of scopes and presents the options to allow or to deny access.
- **Disabled** - Does not display the OAuth Authorization page.

Scopes

Specifies the scopes for which authorization is requested. If no scopes are specified here, the scopes configured in APM for the client application are used.

Customization

Customize the messages that display on the OAuth authorization page when **Prompt for Authorization** is set to **Enabled**:

- **Authorize Message** Specifies the initial wording for the prompt.
- **Scope Message** Specifies the wording that precedes the list of scopes that are specified in the Scopes Assign area of this screen.
- **Allow Message** Provides the label for the button that allows access.
- **Deny Message** Provides the label for the button that denies access.

About OAuth Client

An OAuth Client agent is a policy item that requests authorization and tokens from an OAuth server. An OAuth Client can also validate tokens and get scope data on a per-request basis. The OAuth Client agent provides these configuration elements and options:

Server

Specifies the OAuth server to which this OAuth client directs requests.

Grant Type

Specifies the type of grant that the OAuth client uses.

- **Authorization code** - The client redirects the resource owner to the OAuth server to request an authorization code.
- **Password** - The client uses resource owner password credentials to request an access token from the OAuth server.

Authentication Redirect Request

Specifies an auth-redirect-request type request, which redirects a user to an OAuth server. Displays when **Grant Type** is set to **Authorization code**.

Token Request

Specifies a token-request type of request.

Refresh Token Request

Specifies a token-refresh-request type of request. APM uses this request on a per-request basis.

Validate Token Request

Specifies a validation-scopes-request type of request to validate a token and to get a list of scopes associated with a token. APM uses this request on a per-request basis.

Redirection URI

Specifies the URI for the OAuth server to redirect a user back to the OAuth client. Displays when **Grant Type** is set to **Authorization code**.

Scope

Specifies one or more strings separated by spaces; for example `contacts photo email`. The strings are defined by the OAuth authorization server. Your best source of information for the strings that a particular OAuth authorization server defines could be APIs for OAuth 2.0 scopes on developer sites for OAuth providers.

For the **Authorization code** grant type, an OAuth authorization server prompts the user to grant or deny access to the scopes. For the **Password** grant type, an OAuth authorization server grants permission to the requested scopes based on the user providing resource owner password credentials.

***Note:** Requests are configured in the **Access > Federation > OAuth Client / Resource Server > Requests** area of the product.*

About OAuth Scope

The OAuth Scope agent makes requests to an OAuth authorization server to get scopes associated with a token and to get scope data, such as a user's email address or contact list. The OAuth Scope item provides these elements and options:

Server

Specifies an OAuth server. (OAuth servers in resource server, or client and resource server modes are available for selection.)

Scopes Request

Specifies a validation-scopes-request type request. This request type retrieves a list of scopes associated with the token.

There can be multiple scope data requests in this agent with these elements:

Scope Name

Specifies the name of a scope for which you are requesting data. (The external OAuth provider specifies the names of the scopes that it supports.)

Request

Specifies a scope-data-request type request. This is optional. If the provider does not require this type of request to obtain additional information from an authorization server, you do not need to fill in this field.

***Note:** Requests are configured in the **Access > Federation > OAuth Client / Resource Server > Requests** area of the product.*

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. Typically, an OCSP Auth action follows an action that receives an X.509 certificate. Either a Client Cert Inspection or On-Demand Cert Auth action can receive the X.509 certificate from a user. Either action populates session variables with data that OCSP Auth uses. Similarly, a Machine Cert Auth action can receive an X.509 certificate from a machine and populate session variables.

***Note:** A CRLDP Auth action is valid for use in a per-request policy subroutine when placed after an On-Demand Cert Auth action.*

An OCSP Auth action provides these configuration elements and options:

OCSP Responder

Specifies the OCSP Responder AAA configuration object, defined in the Access Policy AAA servers area of the Configuration utility.

Certificate Type

Specifies the expected type of certificate: **User** or **Machine**.

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.*

***Note:** On-demand certificate authentication does not work when added to a subroutine for a per-request policy that is part of a forward proxy configuration.*

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 > Authentication** 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.

***Note:** When configured in a per-request policy subroutine, some screen elements and options described here might not be available.*

AAA Server

Specifies the RADIUS accounting server; servers are defined in the **Access > Authentication** 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.

***Note:** For a per-request policy subroutine, equivalent functionality is supported through subroutine settings.*

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 > Authentication** 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.

***Note:** For a per-request policy subroutine, equivalent functionality is supported through subroutine settings.*

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 > Authentication** 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 > Authentication** 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.

***Note:** For a per-request policy subroutine, equivalent functionality is supported through subroutine settings.*

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.

Per-Request Policy Item Reference

About per-request policy items

When configuring a per-request policy, a few access policy items are available for inclusion in the policy. Most per-request policy items are unique to a per-request policy.

About Protocol Lookup

A Protocol Lookup item determines whether the protocol of the request is HTTP or HTTPS. It provides two default branches: HTTPS and fallback. Use the Protocol Lookup item early in a per-request policy to process HTTPS traffic before processing HTTP traffic.

About SSL Bypass Set

The SSL Bypass Set item provides a read-only element, **Action**, that specifies the **Bypass** option.

***Note:** For an SSL Bypass Set item to be effective, the client and server SSL profiles on the virtual server must enable SSL forward proxy and SSL forward proxy bypass; the client SSL profile must set the default bypass action to **Intercept**; and the SSL Bypass Set item must occur in the policy before any items that process HTTP traffic.*

About AD Group Lookup

An AD Group Lookup item can branch based on Active Directory group. The item provides one default advanced branch rule expression, `expr { [mcget {session.ad.last.attr.primaryGroupID}] == 100 }`, as an example.

A branch rule expression can include any populated session variable, such as `session.ad.last.attr.primaryGroupID`, `session.ad.last.attr.memberOf`, `session.ad.last.attr.lastLogon`, `session.ad.last.attr.groupType`, `session.ad.last.attr.member`, and so on. As an example, `expr { [mcget {session.ad.last.attr.memberOf}] contains "CN=Administrators" }` is a valid expression.

***Note:** An AD Query action in the access policy can populate the session variables.*

About LDAP Group Lookup

An LDAP Group Lookup item compares a specified string against the `session.ldap.last.attr.memberOf` session variable. The specified string is configurable in a branch rule. The default simple branch rule expression is `User is a member of CN=MY_GROUP, CN=USERS, CN=MY_DOMAIN`; the values `MY_GROUP`, `USERS`, `MY_DOMAIN`, must be replaced with values used in the LDAP group configuration at the user site.

***Note:** An LDAP Query action is required in the access policy to populate the session variable.*

About LocalDB Group Lookup

A per-request policy LocalDB Group Lookup item compares a specified string against a specified session variable.

The string is specified in a branch rule of the LocalDB Group Lookup item. The default simple branch rule expression is **User is a member of** `MY_GROUP`. The default advanced rule expression is `expr { [mcget {session.localdb.groups}] contains "MY_GROUP" }`. In either the simple or the advanced rule, the variable, `MY_GROUP`, must be replaced with a valid group name.

The session variable must initially be specified and populated by a Local Database action in the access policy. A Local Database action reads groups from a local database instance into a user-specified session variable. It can be `session.localdb.groups` (used by default in the LocalDB Group Lookup advanced rule expression) or any other name. The same session variable name must be used in the Local Database action and the LocalDB Group Lookup advanced rule expression.

About RADIUS Class Lookup

The RADIUS Class Lookup access policy item compares a user-specified class name against the `session.radius.last.attr.class` session variable. The specified class name is configurable in a branch rule.

The default simple branch rule expression is **RADIUS Class attribute contains** `MY_CLASS`. The variable `MY_CLASS` must be replaced with the name of an actual class.

***Note:** A RADIUS Acct or RADIUS Auth action is required in the access policy to populate the session variable.*

About Dynamic Date Time

The Dynamic Date Time action enables branching based on the day, date, or time on the server. It provides two default branch rules:

Weekend

Defined as Saturday and Sunday.

Business Hours

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

The Dynamic 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 SSL Intercept Set

The SSL Intercept Set item provides a read-only element, **Action**, that specifies the **Intercept** option.

Note: For an SSL Intercept Set item to be effective, the client and server SSL profiles on the virtual server must enable SSL forward proxy and SSL forward proxy bypass; the client SSL profile must set the default bypass action to **Intercept**; and the SSL Intercept Set item must occur in the policy before any items that process HTTP traffic.

About the Logging action

The Logging action can be used in an access policy or in a per-request policy. In an access policy, the Logging action adds logging for session variables to the access policy. In a per-request policy, the Logging action can add logging for both session variables and per flow variables to the per-request policy.

This action is useful for tracing the 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

For an access policy, specifies text to add to the log file. For a per-request policy, specifies the message text and the session and per-flow variables to add to the message. Complete variable names must be typed. Wildcards are not supported for per-request policies. An example log message for a per-request policy follows.

```
The system found this URL %{perflow.category_lookup.result.url} in these categories %
{perflow.category_lookup.result.categories} and placed it into this category %
{perflow.category_lookup.result.primarycategory}.
```

```
An HTTPS request was made to this host %{perflow.category_lookup.result.hostname}; the per-
request policy set SSL bypass to %{perflow.ssl_bypass_set}.
```

```
Requests from this platform %{session.client.platform} were made during this session %
{perflow.session.id}.
```

Session Variables

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

Note: This option is available only when adding the Logging action to an access policy.

About Category Lookup

A Category Lookup item looks up URL categories for a request and obtains a web response page.

The Category Lookup item provides these elements and options.

Categorization Input

The list specifies these options:

- **Use HTTP URI (cannot be used for SSL Bypass decisions):** For HTTP traffic, this option specifies performing a URL-based lookup. When selected, on a BIG-IP® system with an SWG subscription the **SafeSearch Mode** setting displays.
- **Use SNI in Client Hello (if SNI is not available, use Subject.CN):** For HTTPS traffic, this option specifies performing a host-based lookup.
- **Use Subject.CN in Server Cert:** For HTTPS traffic, this option specifies performing a host-based lookup. (This option is not for use in a reverse proxy configuration.)

SafeSearch Mode

The options are **Enabled** (default) and **Disabled**. When enabled, SWG enables Safe Search for supported search engines.

***Note:** SafeSearch is available only with an SWG subscription.*

Category Lookup Type

Select the category types in which to search for the requested URL. On a BIG-IP® system with an SWG subscription, options are:

- **Select one from Custom categories first, then standard categories if not found**
- **Always process full list of both custom and standard categories**
- **Process standard categories only**

On a BIG-IP® system without an SWG subscription, the available option is **Process custom categories only**. Depending on the selection, the Category Lookup Type item looks through custom categories or standard categories or both, and compiles a list of one or more categories from them. The list is available for subsequent processing by the URL Filter Assign item.

Reset on Failure

When enabled, specifies that SWG send a TCP reset to the client in the event of a server failure.

About Response Analytics

A Response Analytics item inspects a web response page for malicious embedded contents. Response Analytics must be preceded by a Category Lookup item because it obtains a web response page.

***Note:** Response Analytics works only on a BIG-IP® system with an SWG subscription.*

Response Analytics provides these elements and options.

Max Buffer Size

Specifies the maximum amount of response data (in bytes) to collect before sending it for content scanning. The system sends the content for analysis when the buffer reaches this size or when the buffer contains all of the response content. Otherwise, the system retains the response data in the buffer.

Max Buffer Time

Specifies the maximum amount of time (in seconds) for buffering and analyzing response data. If the time elapses at any point in this process, the agent sets the `perflow.response_analytics.failure` variable to 1 (which indicates an ANTserver failure) and discards the response data.

Reset on Failure

When enabled, specifies that SWG send a TCP reset to the client in the event of an ANTserver failure. If disabled and an ANTserver failure occurs, SWG logs all perflow variables and provides the SWG block page to the client.

Exclude Types

Specifies one entry for each type of content to be excluded from content analysis. Images, the **All-Images** type, do not get analyzed.

About Request Analytics

A Request Analytics item inspects an outgoing web request for malicious embedded contents. In a per-request policy, a Request Analytics item must be preceded by a Category Lookup item and followed by a URL Filter Assign item. To block outgoing traffic from chat applications, a Request Analytics item is required.

***Note:** Request Analytics works only on a BIG-IP® system with an SWG subscription.*

Request Analytics provides these elements and options.

Max Buffer Size

Specifies the maximum amount of request data (in bytes) to collect before sending it for content scanning. The system sends the content for analysis when the buffer reaches this size or when the buffer contains all of the request content. Otherwise, the system retains the request data in the buffer.

Max Buffer Time

Specifies the maximum amount of time (in seconds) for buffering and analyzing request data. If the time elapses at any point in this process, the agent sets the `perflow.request_analytics.failure` variable to 1 (which indicates an ANTserver failure) and discards the request data.

Reset on Failure

When enabled, specifies that SWG send a TCP reset to the client in the event of an ANTserver failure. If disabled and an ANTserver failure occurs, SWG logs all perflow variables and provides the SWG block page to the client.

About URL Filter Assign

A URL Filter Assign item looks up the URL filter action for each category that the Category Lookup item found for a request. If any filter action is set to Block, the request is blocked. In a configuration with an SWG subscription, the URL Filter Assign item also uses the analysis from the Response Analytics item, if used, to determine whether to block the request.

By default, the URL Filter Assign item has three branches: Allow, Confirm, and fallback. If the request is not blocked and any filter action is set to Confirm, the per-request policy takes the Confirm branch.

A URL Filter Assign item provides the **URL Filter** element, with a list of filters from which to select.

***Note:** A Category Lookup item must precede the URL Filter Assign item.*

About Application Lookup

An Application Lookup item obtains the name of the application that is being requested and looks up the application family that matches it. By default, this item has a fallback branch only.

Application Lookup can be used to branch by application family or by application name; branch rules are required to do this. If an Application Filter Assign item is included in the per-request policy, an Application Lookup must complete before it.

About Application Filter Assign

An Application Filter Assign item matches an application or application family against an application filter. Application Filter Assign provides one configuration element. The **Application Filter** element specifies the application filter to use in determining whether to block access to an application or allow it. The Application Filter Assign item exits on the Allow branch if the filter action specifies allow. Otherwise, Application Filter Assign exits on the fallback branch.

Important: To supply input for the Application Filter Assign agent, an Application Lookup item must run in the per-request policy sometime prior to it.

About HTTP Headers

An HTTP Headers action supports modifying an outgoing HTTP request to a back-end server. The action supports manipulation of HTTP and cookie headers being sent to back-end servers.

Important: The HTTP Headers item cannot manipulate HTTP cookies in outgoing HTTP requests to any portal access application.

The HTTP Headers item provides these configuration options and elements.

An entry in the HTTP Header Modify table includes these elements.

Header Operation

Specifies **insert**, **append**, **replace**, or **remove**.

Header Name

Specifies the header name on which to operate.

Header Value

Specifies the value on which to operate.

Note: Any per-flow or session variable can be used as a header value, for example, `%{session.user.clientip}` or `%{perflow.session.id}`.

Header Delimiter

Specifies the separator to use when appending a header.

An entry in the HTTP Cookie Modify table includes these elements.

Cookie Operation

Specifies **update** or **delete**.

Note: When **update** is selected and a cookie that matches the name and value does not exist, HTTP Header adds the specified cookie.

Cookie Name

Specifies the name to match.

Cookie Value

Specifies the value to match when deleting a cookie or the new value to set when updating a cookie.

Note: Any per-flow or session variable can be used as a cookie value.

About Select SSO Configuration

The Select SSO Configuration agent enables per-request selection of an SSO configuration from these SSO configuration types:

- HTTP Basic
- NTLMv1
- NTLMv2
- Kerberos

The Select SSO Configuration agent provides these configuration elements and options:

SSO Configuration Name

Select an SSO configuration name from the list.

About OAuth Client

An OAuth Client agent is a policy item that requests authorization and tokens from an OAuth server. An OAuth Client can also validate tokens and get scope data on a per-request basis. The OAuth Client agent provides these configuration elements and options:

Server

Specifies the OAuth server to which this OAuth client directs requests.

Grant Type

Specifies the type of grant that the OAuth client uses.

- Authorization code - The client redirects the resource owner to the OAuth server to request an authorization code.
- Password - The client uses resource owner password credentials to request an access token from the OAuth server.

Authentication Redirect Request

Specifies an auth-redirect-request type request, which redirects a user to an OAuth server. Displays when **Grant Type** is set to **Authorization code**.

Token Request

Specifies a token-request type of request.

Refresh Token Request

Specifies a token-refresh-request type of request. APM uses this request on a per-request basis.

Validate Token Request

Specifies a validation-scopes-request type of request to validate a token and to get a list of scopes associated with a token. APM uses this request on a per-request basis.

Redirection URI

Specifies the URI for the OAuth server to redirect a user back to the OAuth client. Displays when **Grant Type** is set to **Authorization code**.

Scope

Specifies one or more strings separated by spaces; for example `contacts photo email`. The strings are defined by the OAuth authorization server. Your best source of information for the strings that a particular OAuth authorization server defines could be APIs for OAuth 2.0 scopes on developer sites for OAuth providers.

For the **Authorization code** grant type, an OAuth authorization server prompts the user to grant or deny access to the scopes. For the **Password** grant type, an OAuth authorization server grants permission to the requested scopes based on the user providing resource owner password credentials.

***Note:** Requests are configured in the **Access > Federation > OAuth Client / Resource Server > Requests** area of the product.*

About OAuth Scope

The OAuth Scope agent makes requests to an OAuth authorization server to get scopes associated with a token and to get scope data, such as a user's email address or contact list. The OAuth Scope item provides these elements and options:

Server

Specifies an OAuth server. (OAuth servers in resource server, or client and resource server modes are available for selection.)

Scopes Request

Specifies a validation-scopes-request type request. This request type retrieves a list of scopes associated with the token.

There can be multiple scope data requests in this agent with these elements:

Scope Name

Specifies the name of a scope for which you are requesting data. (The external OAuth provider specifies the names of the scopes that it supports.)

Request

Specifies a scope-data-request type request. This is optional. If the provider does not require this type of request to obtain additional information from an authorization server, you do not need to fill in this field.

***Note:** Requests are configured in the **Access > Federation > OAuth Client / Resource Server > Requests** area of the product.*

About per-request policy subroutine items

When configuring a per-request policy subroutine, a few access policy items are available for inclusion in the subroutine. A Confirm Box action (for use with Secure Web Gateway forward proxy configurations) is unique to a per-request policy subroutine.

Per-session policy and subroutine agent differences

The agents in this table are available to per-session policies and to per-request policy subroutines. In a per-request policy subroutine, not all options for an agent are supported and support for some options is implemented differently.

Table 3: Per-Request Policy Subroutine Agents with Differences

Agent	Description
HTTP 401 Response	Supports no authentication or HTTP Basic authentication only.

Agent	Description
Logon Page	A Subsession Variable field replaces the Session Variable field. Split domain from full Username and CAPTCHA Configuration fields do not display because the functionalities are not supported.
AD Auth	Support for multiple logon attempts can be implemented using a macro loop. The Max Logon Attempts Allowed property does not display. The Show Extended Error property is not supported.
LDAP Auth	Support for multiple logon attempts can be implemented using a macro loop. The Max Logon Attempts Allowed property does not display. The Show Extended Error property is not supported.
RADIUS Auth	Support for multiple logon attempts can be implemented using a macro loop. The Max Logon Attempts Allowed property does not display. The Show Extended Error property is not supported.

About AD Auth

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

Note: When configured in a per-request policy subroutine, some screen elements and options described here might not be available.

Type

Specifies Authentication, the type of this Active Directory action.

Server

Specifies an Active Directory server; servers are defined in the **Access > Authentication** 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.

***Note:** For a per-request policy subroutine, equivalent functionality is supported through subroutine settings.*

Max Password Reset Attempts Allowed

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

About Confirm Box

A Confirm Box action presents links for these options: **Continue** and **Cancel**. The action is available for a per-request policy subroutine only and is for use in a Secure Web Gateway (SWG) configuration. Confirm Box offers these elements and options for customization.

Language

Specifies the language to use to customize the Confirm Box 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.*

Message

Specifies the message to display.

Field 1 image

Specifies the icon (red, green, or none) to display with the **Continue** option.

Continue

Specifies the text to display for this option.

Field 2 image

Specifies the icon (red, green, or none) to display with the **Cancel** option.

Cancel

Specifies the text to display for this option.

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 > Authentication** area of the Configuration utility.

***Note:** A CRLDP Auth action is valid for use in a per-request policy subroutine when placed after an On-Demand Cert Auth action.*

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 > Authentication** area of the Configuration utility.

About HTTP 401 Response

The HTTP 401 Response action sends an HTTP 401 Authorization Required Response page to capture HTTP Basic or Negotiate authentication.

***Note:** For a per-request policy subroutine, HTTP 401 Response supports HTTP Basic authentication only.*

The HTTP 401 Response action provides up to 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.

Basic Auth Realm

Specifies the authentication realm for use with Basic authentication.

HTTP Auth Level

Specifies the authentication required for the policy.

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

***Note:** This option is not available for a per-request policy subroutine.*

- **basic+negotiate** - specifies either Basic or Kerberos authentication.

***Note:** This option is not available for a per-request policy subroutine.*

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 iRule Event

An iRule Event action adds iRule processing to an access policy or to a per-request policy subroutine 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 or a per-request policy subroutine.

About LDAP Auth

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

***Note:** When configured in a per-request policy subroutine, some screen elements and options described here might not be available.*

Type

Specifies Authentication, the type of this LDAP action.

Server

Specifies an LDAP server; servers are defined in the **Access > Authentication** 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. Parentheses are required around search strings; (`sAmAccountName=%{session.logon.last.username}`)

UserDN

Specifies the Distinguished Name (DN) of the user. The DN can be derived from session variables.

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.

***Note:** For a per-request policy subroutine, equivalent functionality is supported through subroutine settings.*

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.

***Note:** For a per-request policy subroutine, equivalent functionality is supported through subroutine settings.*

About Logon Page

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.

***Note:** When configured in a per-request subroutine, some screen elements and options described here might not be available.*

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 check box.
- **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 (or Subsession 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`.

***Note:** A per-request policy subroutine uses subsession variables in place of session variables.*

Clean Variable

Specifies whether to clear any value from the variable before presenting the logon page to the user; to clean the variable, select **Yes**. Defaults to **No**.

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 to 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 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.*

***Note:** On-demand certificate authentication does not work when added to a subroutine for a per-request policy that is part of a forward proxy configuration.*

About OAuth Client

An OAuth Client agent is a policy item that requests authorization and tokens from an OAuth server. An OAuth Client can also validate tokens and get scope data on a per-request basis. The OAuth Client agent provides these configuration elements and options:

Server

Specifies the OAuth server to which this OAuth client directs requests.

Grant Type

Specifies the type of grant that the OAuth client uses.

- Authorization code - The client redirects the resource owner to the OAuth server to request an authorization code.
- Password - The client uses resource owner password credentials to request an access token from the OAuth server.

Authentication Redirect Request

Specifies an auth-redirect-request type request, which redirects a user to an OAuth server. Displays when **Grant Type** is set to **Authorization code**.

Token Request

Specifies a token-request type of request.

Refresh Token Request

Specifies a token-refresh-request type of request. APM uses this request on a per-request basis.

Validate Token Request

Specifies a validation-scopes-request type of request to validate a token and to get a list of scopes associated with a token. APM uses this request on a per-request basis.

Redirection URI

Specifies the URI for the OAuth server to redirect a user back to the OAuth client. Displays when **Grant Type** is set to **Authorization code**.

Scope

Specifies one or more strings separated by spaces; for example `contacts photo email`. The strings are defined by the OAuth authorization server. Your best source of information for the strings that a particular OAuth authorization server defines could be APIs for OAuth 2.0 scopes on developer sites for OAuth providers.

For the **Authorization code** grant type, an OAuth authorization server prompts the user to grant or deny access to the scopes. For the **Password** grant type, an OAuth authorization server grants permission to the requested scopes based on the user providing resource owner password credentials.

***Note:** Requests are configured in the **Access > Federation > OAuth Client / Resource Server > Requests** area of the product.*

About OAuth Scope

The OAuth Scope agent makes requests to an OAuth authorization server to get scopes associated with a token and to get scope data, such as a user's email address or contact list. The OAuth Scope item provides these elements and options:

Server

Specifies an OAuth server. (OAuth servers in resource server, or client and resource server modes are available for selection.)

Scopes Request

Specifies a validation-scopes-request type request. This request type retrieves a list of scopes associated with the token.

There can be multiple scope data requests in this agent with these elements:

Scope Name

Specifies the name of a scope for which you are requesting data. (The external OAuth provider specifies the names of the scopes that it supports.)

Request

Specifies a scope-data-request type request. This is optional. If the provider does not require this type of request to obtain additional information from an authorization server, you do not need to fill in this field.

***Note:** Requests are configured in the **Access > Federation > OAuth Client / Resource Server > Requests** area of the product.*

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. Typically, an OCSP Auth action follows an action that receives an X.509 certificate. Either a Client Cert Inspection or On-Demand Cert Auth action can receive the X.509 certificate from a user. Either action populates session variables with data that OCSP Auth uses. Similarly, a Machine Cert Auth action can receive an X.509 certificate from a machine and populate session variables.

***Note:** A CRLDP Auth action is valid for use in a per-request policy subroutine when placed after an On-Demand Cert Auth action.*

An OCSP Auth action provides these configuration elements and options:

OCSP Responder

Specifies the OCSP Responder AAA configuration object, defined in the Access Policy AAA servers area of the Configuration utility.

Certificate Type

Specifies the expected type of certificate: **User** or **Machine**.

About Proxy Select

The Proxy Select agent is for use in selecting the next hop in forward proxy chaining. The Proxy Select agent provides these elements and options:

Pool

Specifies a pool of one or more proxy servers from which to select the next hop. All proxy servers in the pool that you select must support the forward proxy mode that you specify in the **Upstream Proxy Mode** setting.

Upstream Proxy Mode

Specifies whether the next hop is to a forward proxy server that supports **Explicit** forward proxy or **Transparent** forward proxy.

Username

Specifies the name of a user account on the proxy server. To use static credentials to authenticate the user at the next hop, provide the username and password .

Password

Specifies the password for the user account on the proxy server.

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.

***Note:** When configured in a per-request policy subroutine, some screen elements and options described here might not be available.*

AAA Server

Specifies the RADIUS accounting server; servers are defined in the **Access > Authentication** 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.

***Note:** For a per-request policy subroutine, equivalent functionality is supported through subroutine settings.*

About Select SSO Configuration

The Select SSO Configuration agent enables per-request selection of an SSO configuration from these SSO configuration types:

- HTTP Basic
- NTLMv1
- NTLMv2
- Kerberos

The Select SSO Configuration agent provides these configuration elements and options:

SSO Configuration Name

Select an SSO configuration name from the list.

About URL Branching

The URL Branching action is useful for treating a few URLs differently from others. The action provides an Allow branch and a fallback branch. The URL Branching action provides these conditions for defining branch rules.

Equals

The URL must exactly match the specified URL.

Substring

The URL must contain the specified string.

Prefix Match

The URL must start with the specified string.

Suffix Match

The URL must end with the specified string.

Glob match

The URL must match the specified globbing pattern. These globbing patterns are supported:

- * Matches any number of characters (none or one or more).
- ? Matches a single character in these sets: [a-z] or [0-9] or [A-Za-z].
- [characters] Matches one of the specified characters.

- `[^characters]` Matches any characters except for those specified.
- `[!characters]` Matches any characters except for those specified.

Note: To match many URLs, you might consider configuring URL categories in the *Access > Secure Web Gateway* area of the product and using *Category Lookup* in your per-request policy.

About per-request policy endings

An ending provides a result for a per-request policy branch. An ending for a per-request policy branch is one of two types.

Allow

Allows the user to continue to the requested URL.

Reject

Blocks the user from continuing and triggers the access profile Logout screen.

Session Variables

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 (.).

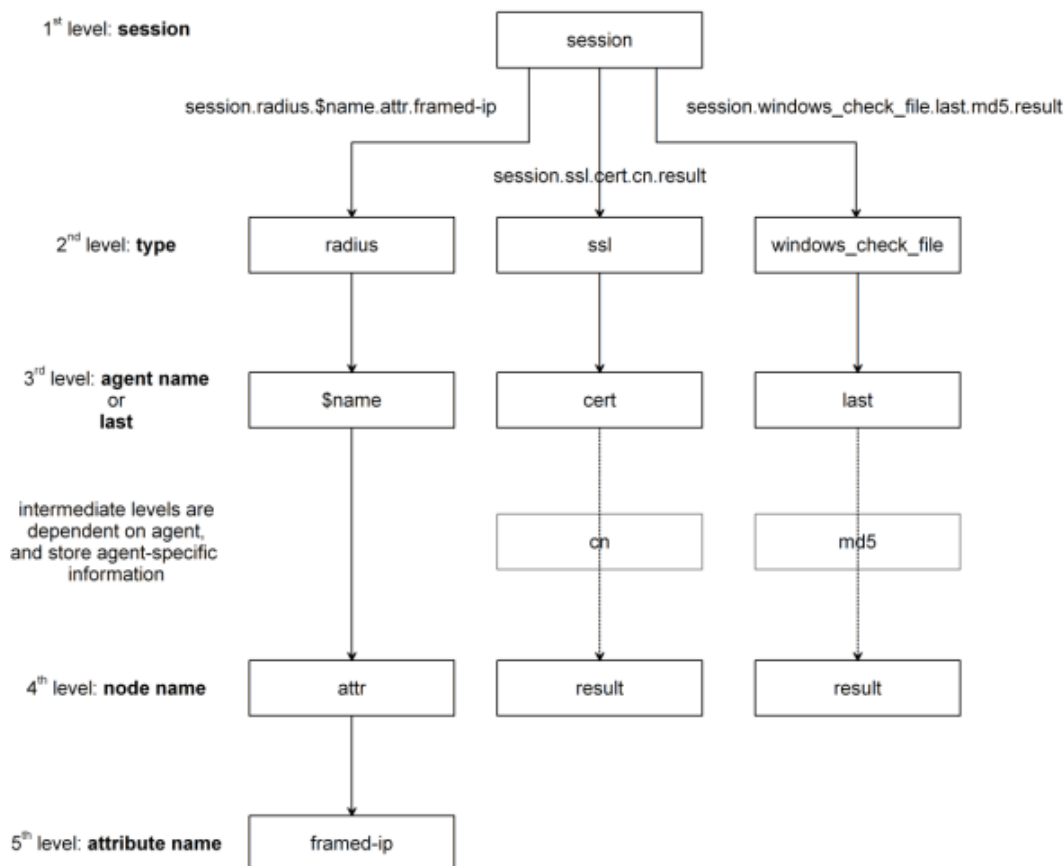


Figure 11: How APM constructs session variable names

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 <code>access_denied</code> .
Redirect Ending	<code>session.policy.result</code>	string	Access policy result: the access policy ended at Redirect. The value is <code>redirect</code> .
	<code>session.policy.result.redirect.url</code>	string	URL specified in the redirect, for example, <code>http://www.siterequest.com</code> .
Allowed Ending	<code>session.policy.result</code>	string	Access policy result: the access policy ended at Allow. The value is <code>allowed</code> .
	<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: <code>network_access</code> or <code>web_application</code> .
Session management	<code>session.ui.mode</code>	enum	<p>UI mode, as determined by HTTP headers. UI mode reflects the protocol that the client used to communicate with the server during APM session establishment and access policy execution. UI mode does not directly map to client type (<code>session.client.type</code>). For example, when BIG-IP Edge Client uses a web browser component to establish a session, the <code>session.ui.mode</code> is set to 0 (Full Browser). Values:</p> <ul style="list-style-type: none"> • 0 - Full Browser • 6 - Pocket PC (browser) • 7 - Standalone Client (clientless mode, no support for endpoint inspection; not Edge Client) • 8 - ActiveSync Client

Action Item	Session Variable	Type	Description
			<ul style="list-style-type: none"> 9 - Mobile Browser (smart phone) 10 - Citrix Receiver
	<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 or "Standalone" (Edge Client).
	<code>session.client.version</code>	string	
	<code>session.client.jailbreak</code>	bool	Mobile device is jailbroken/rooted: <ul style="list-style-type: none"> 0 - No 1 - Yes
	<code>session.client.js</code>	bool	Client is capable of executing JavaScript: <ul style="list-style-type: none"> 0 - No 1 - Yes
	<code>session.clientactivex</code>	bool	Client is capable of running ActiveX Controls: <ul style="list-style-type: none"> 0 - No 1 - Yes
	<code>session.client.plugin</code>	bool	
	<code>session.client.platform</code>	string	Client platform as determined by HTTP headers: <ul style="list-style-type: none"> "Android" "ChromeOS" "iOS" "Linux" "MacOS" "Win10" "Win2k" "Win2k" "Win7" "Win8.1" "Win8" "WindowsPhone" "WinLH" "WinNT" "WinVI" "WinXP"
	<code>session.user.access_mode</code>	string	Enables direct access to a Citrix resource from the webtop. Example: local.

Action Item	Session Variable	Type	Description
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
	<code>session.ssl.cert.end</code>	string	Validity end date
	<code>session.ssl.cert.start</code>	string	Validity start date
	<code>session.ssl.cert.issuer</code>	string	Certificate issuer
	<code>session.ssl.cert.whole</code>	string	The whole certificate
	<code>session.decision_box.last.result</code>	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	<code>session.windows_check_file.\$name.item_0.exist</code>	string	True - if all files exist on the client.

Action Item	Session Variable	Type	Description
	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	<p>A bitmask used when CAPTCHA is enabled.</p> <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 <p><i>Note: Should not be used by external modules because it is intended for very specific purposes.</i></p>
Machine Cert Auth	session.check_machinecert.last.result	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

Action Item	Session Variable	Type	Description
			<p>format. Incorrect configuration. (For example, CA profile is not configured). Linux client is trying to access the agent.</p> <hr/> <p>Note: The Machine Cert Auth action is not supported on Linux.</p> <hr/>
OTP Generate	session.otp.assigned.val	string	Generated one-time password value to send to the end user. Example message: One-Time Passcode: %{session.otp.assigned.val}
	session.otp.assigned.expire	string	Internally used timestamp; OTP expiration in seconds since this date and time: (00:00:00 UTC, January 1, 1970)
	session.otp.assigned.ttl	string	OTP time-to-live; configurable as OTP timeout in seconds. Example message: OTP expires after use or in %{session.otp.assigned.ttl} seconds
OTP Verify	session.otp.verify.last.authresult	bool	<p>0 or 1.</p> <ul style="list-style-type: none"> 0 - OTP authentication failed 1 - OTP authentication passed
RADIUS action	session.radius.\$name.authresult	bool	<p>0 or 1.</p> <ul style="list-style-type: none"> 0 - RADIUS authentication failed 1 - RADIUS authentication passed
	session.radius.\$name.attr.\$attr_name	string	User attributes retrieved during RADIUS authentication. Each attribute is converted to a separate session variable.
Resource allocation	session.assigned.resources.at	string	Space-delimited list of names of assigned App tunnel resources.
	session.assigned.resources.na	string	Space-delimited list of names of assigned Network Access resources.
	session.assigned.resources.pa	string	Space-delimited list of names of assigned Portal Access resources.
	session.assigned.resources.rd	string	Space-delimited list of names of assigned remote desktop resources.
	session.assigned.resources.saml	string	Space-delimited list of names of assigned SAML resources.
	session.assigned.webtop	string	Name of the assigned webtop.
Windows Info	session.windows_info_os.\$name.ie_version	string	Stores the Internet Explorer version

Action Item	Session Variable	Type	Description
	session.windows_info_os. \$name.ie_updates	string	List of installed SP and KB fixes for Internet Explorer. For example: " SP2 KB12345 KB54321 "
	session.windows_info_os. \$name.platform	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
	session.windows_info_os. \$name.updates	string	List of installed SP and KB fixes for Windows. For example, " SP2 KB12345 KB54321 "
	session.windows_info_os.\$name.user	string	List of current Windows user names
	session.windows_info_os. \$name.computer	string	List of computer names
Windows Process	session.windows_check_process. \$name.result	integer	0, 1, or -1. <ul style="list-style-type: none"> 0 - Failure 1 - Success -1 - Invalid check expression
Windows Registry	session.windows_check_registrys. \$name.result	integer	0, 1, or -1. <ul style="list-style-type: none"> 0 - Failure 1 - Success -1 - Invalid check expression

sessiondump command usage

The `sessiondump` command syntax includes one operation and one or more arguments and flags.

Usage

`sessiondump <operation> <arguments> <flags>`

Table 4: Operation

Name	Description
help	Show this help message
list	Show list of all sessions
allkeys	Show all session variables for all sessions
locks	Show list of session locks
ip	Show list of IP to session maps
ntlm	Show list of NTLM credentials to session maps

Table 5: Arguments

Name	Description
sid	Show all session variables for a session
delete	Delete a specific session
lockdelete	Delete all or a specific session lock

Table 6: Flags

Name	Description
savetofile	Save all results to a file
hidden	
debug	

Perflow Variables

Per-flow and subsession variables

Per-flow variables exist only while a per-request policy runs. Per-flow variables for a per-request policy subroutine exist while the subsession exists. Multiple subsessions can run simultaneously. The table lists per-flow variables and their values.

Per-flow and subsession variables

Per-flow variables exist only while a per-request policy runs. Per-flow variables for a per-request policy subroutine exist while the subsession exists. Multiple subsessions can run simultaneously. The table lists per-flow variables and their values.

Name	Value
perflow.agent_ending.result	0 (success) or 1 (failure).
perflow.application_lookup.result.families	Comma-separated list of application families.
perflow.application_filter_lookup.result.action	0 (reject) or 1 (allow).
perflow.application_lookup.result.effective_application	Name of the application that is ultimately used.
perflow.application_lookup.result.effective_family	Name of the application family that is ultimately used.
perflow.application_lookup.result.names	Comma-separated list of application names.
perflow.application_lookup.result.primary_application	Name of the application that APM® determines is the primary one.
perflow.application_lookup.result.primary_family	Name of the application family that Access Policy Manager® (APM) determines is the primary one. (An application might fit into more than one application family.)
perflow.bypass_lookup.result.ssl	0 (http) or 1 (https).
perflow.category_lookup.failure	0 (success) or 1 (server failure).
perflow.category_lookup.result.categories	Comma-separated list of categories.
perflow.category_lookup.result.customcategory	Unique number that identifies a custom category; used internally.
perflow.category_lookup.result.effective_category	Name of the category that is ultimately used.
perflow.category_lookup.result.filter_name	Name of the URL filter.
perflow.category_lookup.result.hostname	Host name retrieved from SSL input.
perflow.category_lookup.result.numcategories	Integer. Total number of categories in the comma-separated list of categories.

Name	Value
perflow.category_lookup.result.primarycategory	Name of the category that APM determines is the primary one. (A URL might fit into more than one category, such as news and sports.)
perflow.category_lookup.result.url	Requested URL.
perflow.protocol_lookup.result	http or https. Defaults to https.
perflow.response_analytics.failure	0 (success) or 1 (server failure).
perflow.session.id	Session ID.
perflow.ssl_bypass_set	0 (bypass) or 1 (intercept). SSL Bypass Set and SSL Intercept Set items update this value.
perflow.ssl.bypass_default	0 (bypass) or 1 (intercept). Specified in the client SSL profile, used when SSL Bypass Set and SSL Intercept Set items not included in per-request policy.
perflow.urlfilter_lookup.result.action	0 (reject) or 1 (allow).
perflow.username	User name.
perflow.on_demand_cert.result	0 (success) or 1 (failure) of On-Demand Certificate authentication in the subroutine.
perflow.decision_box.result	0 (continue) or 1 (cancel) selected for the Confirm Box action in the subroutine.
perflow.subroutine.out_terminal	Name of the subroutine out terminal.
perflow.subroutine.invalidated	0 (validated) or 1 (invalidated) subroutine.
perflow.subroutine.loop_countdown	Number of iterations remaining on a subroutine loop.
subsession.logon.last.username	User name for the last login.
subsession.logon.last.authtype	Last authentication type
subsession.ad.last.actualdomain	Domain name for the last login.
subsession.ad.last.authresult	0 (success) or 1 (failure) of Active Directory authentication in the subroutine.
subsession.ad.last.errmsg	Displays the error message for the last login.
subsession.ldap.last.authresult	0 (success) or 1 (failure) of LDAP authentication in the subroutine.
subsession.ldap.last.errmsg	Displays the error message for the last login.
subsession.radius.last.attr.filter-id	RADIUS attribute filter ID
subsession.radius.last.attr.framed-compression	RADIUS attribute framed compression
subsession.radius.last.attr.framed-mtu	RADIUS attribute framed MTU
subsession.radius.last.attr.framed-protocol	RADIUS attribute framed protocol
subsession.radius.last.attr.service-type	RADIUS attribute service type.
subsession.radius.last.errmsg	Displays the error message for the last login.

Name	Value
<code>subsession.radius.last.result</code>	0 (success) or 1 (failure) of RADIUS authentication in the subroutine.

Perflow Variables

Per-Request Policy Maintenance

Customizing messages for the per-request policy Reject ending

You need an access profile configured.

Customize the messages to display when a per-request policy terminates on a Reject ending. When this happens, the per-request policy triggers the access profile Logout screen.

1. On the Main tab, click **Access > Profiles / Policies > Customization > General**.
The Customization tool appears in General Customization view, displaying **Form Factor: Full/Mobile Browser** settings.
2. In the left pane, select the Text tab.
A navigation tree displays in the left pane.
3. Expand the **Access Profiles** folder.
Folders for access profiles that are configured on the BIG-IP® system in the current partition display.
4. Expand the folder for access profile that you want to update.
Folders for access profile objects display.
5. Expand the **Logout** folder for the access profile.
The **General** setting displays in the folder.
6. Click **General**.
Message settings display in the right pane.
7. In the right pane, update values.
8. On the menu bar, click **Save**.
9. Click the **Apply Access Policy** link to apply and activate the changes to the policy.
10. On the list of access profiles to apply, verify that the access profile that you updated is selected.
11. Click the **Apply Access Policy** button.

Exporting and importing a per-request policy across BIG-IP systems

Export a per-request policy from one BIG-IP® system and import it on another (at the same product version level) to copy a policy across systems.

***Note:** Per-request policy import does not support the import of custom categories or the URLs defined for them. Before you import a per-request policy from one BIG-IP system to another BIG-IP system, you must first list any custom categories configured on the source system and make sure you have the same custom categories on the target system. Otherwise, import will fail.*

1. On the Main tab, click **Access > Profiles / Policies > Per-Request Policies**.
The Per-Request Policies screen opens.
2. Click the link in the **Export** column for the policy that you want to export.
A file downloads.
3. Note the list of custom categories:
 - a) Click **Access Policy > Secure Web Gateway > URL Categories**.
 - b) Expand the Custom Categories list.
4. Log in to the Configuration utility on the BIG-IP system where you want to import the per-request policy.

5. Verify that the custom categories that exist on the other BIG-IP system also exist on this BIG-IP system:
 - a) Click **Access Policy > Secure Web Gateway > URL Categories**.
 - b) Expand the Custom Categories list.
 - c) Create any additional custom categories needed to match the list on the other BIG-IP system.

The import process does not add URLs to custom categories. To include the URLs defined for a custom category on the source system, you can add them to the target system now or wait until after you import the per-request policy.
6. On the Main tab, click **Access > Profiles / Policies > Per-Request Policies**.

The Per-Request Policies screen opens.
7. Click **Import**.

An Import Policy screen displays.
8. In **New Policy Name**, type a name.
9. For **Config File Upload**, click **Browse**, locate and select the file downloaded from the other BIG-IP system.
10. To reuse objects already existing on this BIG-IP system, select the **Reuse Existing Objects** check box.
11. Click **Import**.

Tcl Usage

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	<i>mcget</i> 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.

Command or Operator	Type	Description
switch	Rule operator	Evaluates one of several scripts, depending on a given value.
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 expr.
Rule operator	contains	This operator checks for the string PD.
Return values	0 or 1	0 usually indicates failure, while 1 usually indicates success.

Variable Assign Reference

Variable assign: domain plus username example

A Tcl command that separates the values of two session variables with a backslash can provide the value for a domain and user name in this format *DOMAIN\USERNAME*.

Tcl expression

```
[mcget {session.ntlm.last.domain}]\[mcget {session.ntlm.last.username}]
```

The expression obtains the values of the session variables and concatenates them using a backslash as a separator. Both backslashes in the expression are needed to work with Tcl backslash substitution rules.

Tcl expression in Variable Assign entry

The screenshot shows a configuration window with two main sections. The left section, titled 'Custom Variable', has a dropdown menu set to 'Custom Variable' and a radio button labeled 'Unsecure'. Below this is a text area containing the text 'domainuser'. The right section, titled 'Custom Expression', has a dropdown menu set to 'Custom Expression' and a text area containing the Tcl expression: `[mcget {session.ntlm.last.domain}]\[mcget {session.ntlm.last.username}]`. An equals sign '=' is positioned between the two sections.

A Tcl command in the **Custom Expression** pane provides the value for the custom variable, *domainuser*.

Variable assign: text assignment example

Variable assign configuration: a custom variable with a text value

The screenshot shows a configuration window with two main sections. The left section, titled 'Custom Variable', has a dropdown menu set to 'Custom Variable' and a radio button labeled 'Unsecure'. Below this is a text area containing the text 'session.last.logon.last.domain'. The right section, titled 'Text', has a dropdown menu set to 'Text' and a text input field containing the text 'siterequest'. An equals sign '=' is positioned between the two sections.

Predefined session variable attributes

In the Variable Assign action, you can select Predefined Session Variable as the type of variable to assign. Then you can select a predefined variable and assign a value to it. The table lists the variables that you can select and provides the corresponding session variable name and expected format for the session variable.

Variable (selection)	Session variable	Format	Description
Network Access Client IPv4	session.assigned.clientip	IPv4 address, for example 192.168.12.10	Stores the client IPv4 address assigned by Access Policy Manager® (APM®) after the access policy completes.
Network Access Client IPv6	session.requested.ipv6_clientip	IPv6 address, for example 2001:0db8:0000:0000:0000:ff00:0042:8329	Stores the client IPv6 address that APM assigns after the access policy completes.
ACLs	session.assigned.acls	A space-delimited list of assigned ACLs. For example ACL1 ACL3 ACL5.	Stores the assigned static ACLs that APM assigns after the access policy completes.
Inactivity Timeout	session.inactivity_timeout	The inactivity timeout currently assigned to the session. For example 600.	Stores the inactivity timeout that APM assigns to the session after the access policy completes.
Maximum Session Timeout	session.max_session_timeout	The maximum session timeout currently assigned to the session. For example 2000.	Stores the maximum session timeout that APM assigns to the session after the access policy completes.
Network Access Resource	session.assigned.resources.name	A space-delimited list of network access resources currently assigned to the session. For example na1 na2 na3.	Stores the network access resources that APM assigns to the session after the access policy completes.
SNAT	session.assigned.snat.type	The attribute value is 0, 2, or 3. <ul style="list-style-type: none"> 0 - None (no SNAT) 2 - SNAT pool (assigned with the variable <i>snatpool_name</i>) 3 - Automap 	Stores the type of SNAT that APM assigns to the session after the access policy completes.
SNAT Pool	session.assigned.snat.value	The name of the SNAT pool assigned to the session, for example, snat1.	Stores the SNAT pool that APM assigns to the session after the access policy completes.

Variable (selection)	Session variable	Format	Description
Route Domain	session.assigned.route_domain	The name of the route domain assigned to the session, for example, rd1.	Stores the route domain that APM assigns to the session after the access policy completes.
UUID	session.assigned.uuid	The name of the UUID assigned to the session, for example, 550e8400-e29b-41d4-a716-446655440000.	Stores the UUID that APM assigns to the session after the access policy completes.
Webtop	session.assigned.webtop	The name of the webtop assigned to the session, for example, full_webtop.	Stores the webtop that APM assigns to the session after the access policy completes.

Windows Registry Reference

Overview: Policy branching based on Windows Registry values

You can create access policy branches using the values of Windows Registry keys on the client. You can use the GET operator in the Windows Registry action to fetch values from the client. To ensure client security, you must first configure the Windows registry on each client to allow trusted BIG-IP® systems to fetch specific Windows Registry values. Without client configuration, the GET operator fails.

Task summary

Configuring clients for Windows Registry GET operation

Viewing trusted server registry keys and subkeys on a client

Fetching the value of a Windows Registry key from a client

Registry screenshot: Allowed keys for a trusted server

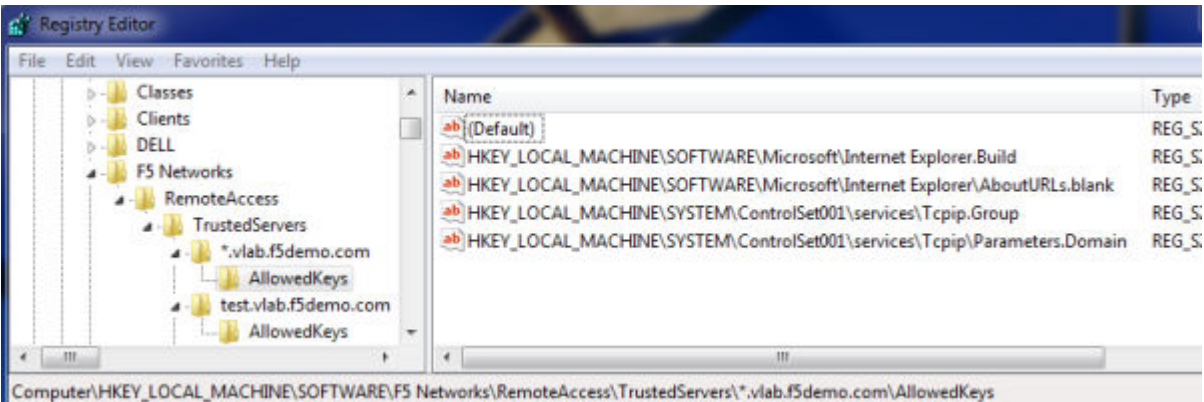


Figure 12: Registry Allowed keys for a trusted server

Example: Allowed registry key value fetched

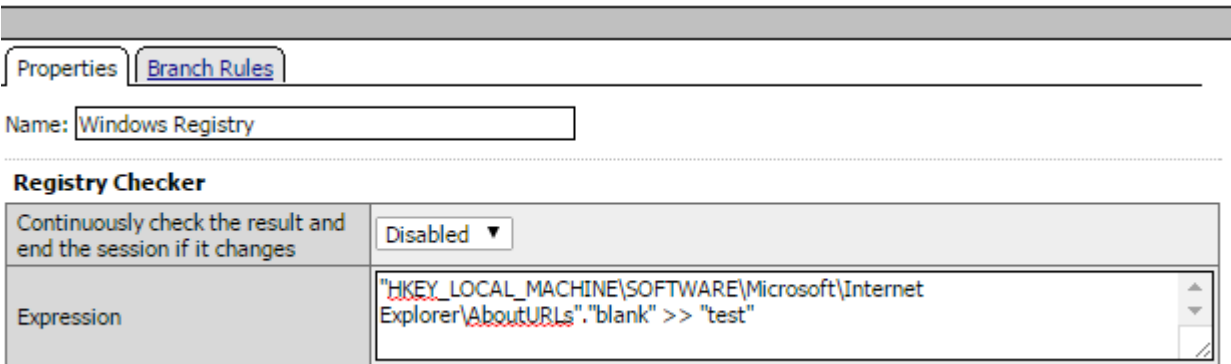


Figure 13: Windows Registry expression in the visual policy editor

The expression uses the GET (>>) operator to fetch the value of the registry key, HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Internet Explorer\AboutURLs".blank", into the user-defined session variable test.

Configuring clients for Windows Registry GET operation

To ensure that only Access Policy Manager® (APM®) can fetch a value from the Windows Registry on a client, you must create registry entries on the client. The entries must specify the BIG-IP® systems that are trusted servers and the specific registry key values that each server is allowed to fetch.

Note: Use Microsoft Group Policy or any other client desktop management system to populate the entries.

1. For the trusted servers, create this registry location: HKEY_LOCAL_MACHINE\Software\F5 Networks\RemoteAccess\TrustedServers.

2. Add subkeys that specify the trusted server locations.

A subkey name can be a fixed server location, such as `www.siterequest.com`, or a regular expression that begins with a wildcard, such as `*.siterequest.com`. The asterisk (*) is the only supported wildcard.

Note: When server names are defined with wildcards, the Windows Registry action selects the most specific server name. For example, for a client configured with these trusted servers: `computer.subd.domain.com`, `*.subd.domain.com`, and `*.domain.com`, Windows Registry prefers: `computer.subd.domain.com` over `*.subd.domain.com` and `*.domain.com`

Here is an example subkey for a trusted server location: HKEY_LOCAL_MACHINE\Software\F5 Networks\RemoteAccess\TrustedServers*.site1.com.

Here is another example subkey: HKEY_LOCAL_MACHINE\Software\F5 Networks\RemoteAccess\TrustedServers\www.site2.com.

3. For each trusted server location, add this subkey: AllowedKeys.

Here is an example: HKEY_LOCAL_MACHINE\Software\F5 Networks\RemoteAccess\TrustedServers*.site1.com\AllowedKeys

Here is another example: HKEY_LOCAL_MACHINE\Software\F5 Networks\RemoteAccess\TrustedServers\www.site2.com\AllowedKeys

4. Add values to each AllowedKeys subkey; populate each value with a specific registry key value that the server is allowed to fetch.

The format for the value is `registry path.value`.

Note: When specifying values, bear in mind that the Windows Registry action supports fetching only these Windows Registry data types: `REG_DWORD`, `REG_SZ`, and `REG_MULTI_SZ`.

Here are two example values:

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\services\Tcpip\Parameters.Domain

HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\services\Tcpip.Group

If the example values exist for the HKEY_LOCAL_MACHINE\Software\F5 Networks\RemoteAccess\TrustedServers*.site1.com\AllowedKeys key, it implies that any server that matches *.site1.com can fetch the value `Domain`, from this registry location HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\services\Tcpip\Parameters.Domain and can fetch the value `Group` from this registry location HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\services\Tcpip.Group.

Viewing trusted server registry keys and subkeys on a client

You can verify the trusted servers and allowed key values on a client by running a diagnostic report using BIG-IP® Edge Client®. If present on the client, the `HKEY_LOCAL_MACHINE\SOFTWARE\F5 Networks\RemoteAccess\TrustedServers` Windows Registry key and its subkeys are included in the report.

***Note:** As an alternative, you can use the Client Troubleshooting Utility to verify trusted server and allowed key values.*

1. Open the BIG-IP® Edge Client® user interface.
On a client with a **Start** button, you can type **BIG-IP** in the search field and, in the results, click **BIG-IP Edge Client**.
2. Click the **View Details** button.
The Details popup screen displays.
3. Click the **Diagnostics Report** button.
A Save As popup screen opens.
4. Select a location, specify a file name, and click **Save**.
A Collecting data popup screen remains open until the report completes.
5. Navigate to the location with the downloaded file, extract the files to a folder, and click the HTML file in the folder.
The F5 Report displays in a browser screen.
6. Scroll down to the **MS Remote Access Diagnostic** section of the table of contents.
7. Look for a link that includes the word **TrustedServers**.
If you do not find such a link, then trusted servers and allowed keys are not configured on the client.
The link in the table of contents should include this path: `HKLM\Software\F5 Networks\RemoteAccess\TrustedServers`.
8. If the link exists, click it to view the subkeys and values configured on the client.

Fetching the value of a Windows Registry key from a client

Before this access policy can run successfully, clients must be configured to allow trusted BIG-IP® systems to fetch specific Windows Registry key values.

You can use a Windows Registry action to fetch values from the Windows Registry on the client.

1. On the Main tab, click **Access > Profiles / Policies**.
The Access Profiles (Per-Session Policies) screen opens.
2. In the Per-Session Policy column, click the **Edit** link for the access profile you want to configure.
The visual policy editor opens the access policy in a separate screen.
3. On a policy branch, click the (+) icon to add an item to the policy.
A popup screen displays actions on tabs, such as General Purpose and Authentication, and provides a search field.
4. Click the Endpoint Security (Client-Side) tab.
5. Select **Windows Registry** and click **Add Item**.
A popup properties screen opens.
6. In the **Expression** field, type an expression that includes these items: the name of a Windows Registry key value, the >> operator, and a name for use as a variable.
The Windows Registry key value used in the expression must match a registry key value that the client allows a trusted server to fetch.

Here is an example expression: "HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\services\Tcpip\Parameters"."Domain" >> "variable_name" where HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\services\Tcpip\Parameters is the registry key, *Domain* is the name of the value to fetch and >> is the GET operator. If GET is successful, then *variable_name* is used to store the value in a session variable formatted like this:
`session.windows_check_registry.last.data.variable_name.`

7. Click Finished.

The popup screen closes.

8. Click Save.

The properties screen closes and the policy displays.

You added an action to fetch a registry key value from the Windows Registry on the client. This is not a complete access policy.

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

Note: To ensure that logging is configured to meet your requirements, verify the log settings for the access profile.

Legal Notices

Legal notices

Publication Date

This document was published on February 13, 2017.

Publication Number

MAN-0507-05

Copyright

Copyright © 2017, 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

For a current list of F5 trademarks and service marks, see <http://www.f5.com/about/guidelines-policies/trademarks>.

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: <https://f5.com/about-us/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.

Legal Notices

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.

Index

A

access

- date-based 55
- time-based 55

access license usage

- checking from an access policy 57

access policy

- about adding a logon page 25, 87
- about localizing a logon page 25, 87
- Active Directory authentication 64, 83
- Active Directory query 65
- adding a blank item 16
- adding a HTTP 407 response 25
- adding a mapping item 21
- adding a preconfigured item 17
- adding a VMware View disclaimer logon page 28
- adding a VMware View SecurID logon page 28
- adding a VMware View Windows logon page 28
- adding an assignment item 20
- adding an external logon page 23
- adding an HTTP 401 response 24, 85
- adding an item with configurable properties 18
- adding connectivity 30
- assigning a variable 33
- assigning a virtual keyboard 27
- assigning a webtop 30, 35
- assigning a webtop link 30
- assigning an ACL 29, 30
- assigning iRule events 61, 86
- assigning resources 32
- assigning webtop links 35
- assigning webtop sections 35
- certificate inspection 66
- configuring date-based access 55
- configuring geolocation-based access 56
- configuring license-based access 57
- configuring time-based access 55
- CRLDP auth action 66, 84
- defining an action 15
- defining an item 15
- defining SSO credential mapping 33
- determining mobile device status 57
- HTTP auth action 66, 69, 85, 91
- IP reputation action 56
- item, adding 8
- Kerberos authentication 66
- LDAP authentication 86
- LDAP query 67
- matching IP subnet 56
- modifying cookies 80
- modifying HTTP headers 80
- per-request policy compared 12
- profile type, related 8
- RADIUS Acct action 71, 72
- RADIUS auth action 71, 91
- requesting proxy authentication 25
- RSA SecurID action 72

access policy (continued)

- SAML authentication 71
- sending email 60
- size limit 12
- TACACS+ accounting action 72
- Transparent Identity Import action 73

access policy branching

- by ActiveSync protocol 52
- by client capable of client-side checks 55
- by client operating system 52
- by client type 53
- by landing URI 57
- by Microsoft Exchange protocol 52

access policy endings

- about 11

access policy item

- defining 15

access policy result

- session variables 96

access policyNTLM authentication result

- checking NTLM authentication result 67
- checking, from an access policy 67

access profile type

- action items, related 8

ACL

- assigning from an access policy 29
- dynamic, about 31
- formats supported 31

ACL assign action

- about 30

Active Directory authentication

- session variables 96

Active Directory queryLDAP authenticationLDAP

- queryRADIUS
- session variables 96

ActiveSync protocol

- detecting on client 52

ActiveX controls

- and client-side actions 35

AD Auth

- and subroutine logon retry 82

AD group lookup

- dependence on access policy 75
- per-request policy item 75

AD group resource assign

- in an access policy 29

Advanced Resource Assign

- session variables 96

Advanced Resource Assign action

- about 30

agent-specific information

- in session variable name 95

Android

- operating system check 52

anti-spyware software check

- in an access policy 35

antivirus software check

- dependency on EPSEC software 36

- antivirus software check (*continued*)
 - in an access policy 36
- application access
 - using a filter to control 80
- application family
 - branching by 79
- Application Filter
 - reliance on Application Lookup 79, 80
- application name
 - branching by 79
- Apply Access Policy 12
- assigning resources
 - to active directory groups 29
 - to LDAP groups 31
- assignment access policy item
 - adding dynamically 20
 - adding to an access policy 20
- assignment items
 - resource assignment 29
 - variable assignment 29
- authentication items
 - AAA servers 63
 - certificate revocation status 63
 - NTLM Auth 63
 - RADIUS accounting 63
 - SSL certificates 63
 - TACACS+ accounting 63

B

- BIG-IP Edge Client
 - client type, detecting 53
- BIG-IP Edge Portal
 - client type, detecting 53
- blank access policy item
 - adding to an access policy 16
- blank item
 - adding dynamically 16
- branch rule
 - advanced example 110
 - Tcl syntax 109
- browser plug-ins
 - and client-side actions 35
- BWC Policy
 - in an access policy 30

C

- category lookup
 - configuring Safe Search 77
 - per-request policy item 77
 - providing content for response analytics 77
- Citrix Receiver
 - client type, detecting 53
- Citrix Receiver (legacy)
 - client type, detecting 53
- client certificate authentication
 - session variables 96
- client certificate inspection
 - client SSL profile, settings for 66
- Client Type action
 - Client OS action, compared with 53

- Client Type action (*continued*)
 - supporting multiple traffic types 53
- Client-Side Capability action
 - about 55
 - client-side checks 55
 - recommendation 55
- concurrent users
 - checking from an access policy 57
- configurable access policy item
 - adding dynamically 18
 - adding to an access policy 18
- confirm box
 - configuring a message 84
 - in a per-request policy subroutine 84
- connectivity license usage
 - checking from an access policy 57
- custom expression
 - Tcl syntax 109

D

- decision box
 - in an access policy 58
 - showing at logon 58
- Decision box
 - session variables 96
- documentation, finding 13
- dynamic date time
 - per-request policy item 76

E

- email
 - adding to an access policy 60
- endings
 - for access policy branches 11
 - for per-request policy branches 93
- endpoint security
 - server-side 52
- endpoint security (client-side)
 - benefits 35
 - purpose 35
- endpoint security (client-side) requirements
 - ActiveX controls 35
 - browser plug-ins 35
 - client component installation 35
- EPSEC software
 - and supported antivirus software 36
 - and supported firewall software 37
 - and supported hard disk encryption software 38
 - and supported Health Agent software 47
 - and supported patch management software 44
 - and supported peer-to-peer software 45
- expression
 - using the GET operator 117
 - Windows Registry example 117
- External Logon page action
 - about 23

F

- File check

File check (*continued*)
 session variables 96

files

checking client for existence of 39, 40, 46
 checking properties of 39, 40, 46
 on client systems 39, 40, 46

firewall software check

dependency on EPSEC software 37
 in an access policy 37

forward proxy chaining

about next hop selection 91
 to a transparent 91
 to an explicit forward proxy 91

G

general purpose items

customizing a log 58
 displaying a message 58
 presenting two choices 58
 processing an iRule 58
 reading from a local database 58
 sending email 58
 writing to a local database 58

guides, finding 13

H

hard disk encryption client-side check
 dependency on EPSEC software 38
 in an access policy 38

HTTP 401 response

about 24, 85

HTTP 401 Response

in per-request policy subroutine 82

HTTP 407 response action

about 25

HTTP header modify

about 80

I

in an access policy 46

iOS

operating system check 52

IP address

matching configuring 56
 matching to physical location 56

IP address intelligence categories

in access policy branch rules 56

IP reputation

in an access policy 56

iRule events

adding to a per-request policy subroutine 61, 86
 adding to an access policy 61, 86

L

LDAP Auth

and subroutine logon retry 82

LDAP group lookup

LDAP group lookup (*continued*)

dependence on access policy 75
 per-request policy item 75

LDAP group resource assign

in an access policy 31

license limits

and per-request policy subroutine 12

license-based access

configuring 57

Linux

operating system check 52

Linux file check

in an access policy 39

Linux process action

in an access policy 39

local database authentication

in an access policy 67, 87

local traffic pool

about assigning 31

local user database, reading

in an access policy 61

local user database, writing

in an access policy 61

LocalDB group lookup

dependence on access policy 75
 per-request policy item 75

localization

customizing messages for per-request policy Reject 107

location-based access 56

logging

in an access policy 62, 77
 per-flow variable 103

logon items 23

Logon Page (CAPTCHA challenge)

session variables 96

logon page action

about 25, 87

logon text

protecting with a virtual keyboard 27

M

Mac

operating system check 52

Mac file check

in an access policy 40

Mac process action

in an access policy 40

Machine Cert Auth

session variables 96

machine cert auth check

in an access policy 41

Machine info action

in an access policy 43

Linux support 43

MAC address 43

Mac support 43

Windows support 43

macro

about 10

adding as an access policy action 9

nesting, limits of 11

- macro (*continued*)
 - terminal, about 10
- macrocall
 - about 10
 - adding to a macro 11
 - adding to access policy 10
- macrocalls
 - on add item screen 9
- manuals, finding 13
- mapping access policy item
 - adding dynamically 21
 - adding to an access policy 21
- maximum logon attempts
 - configuring for a subroutine 82
- mcget command 109
- message box
 - in an access policy 63
 - showing at logon 63
- Microsoft Exchange protocol
 - detecting on client 52
- mobile browser
 - client type, detecting 53
- mobile device
 - determining jailbroken status 57
 - determining rooted status 57

N

- next hop
 - about how to specify 91

O

- OAuth Authorization agent
 - about authorizing a client to access user information 68
- OAuth Client agent
 - about 68, 81, 89
 - defined 68, 81, 89
 - properties of 68, 81, 89
- OAuth Scope agent
 - about 69, 82, 90
 - defined 69, 82, 90
 - properties of 69, 82, 90
- on-demand certificate authentication
 - about 70, 89
 - configuring for iPhone 70, 89
 - configuring for iPod 70, 89
 - Safari, behavior with 70, 89
- operating system check 52
- OTP Generate
 - about 70
 - session variables 96
- OTP Verify
 - about 70
 - session variables 96

P

- patch management software check
 - dependency on EPSEC software 44
 - in an access policy 44
- peer-to-peer software check

- peer-to-peer software check (*continued*)
 - dependency on EPSEC software 45
 - in an access policy 45
- per-request policy
 - access policy compared 12
 - and Secure Web Gateway 12
 - configuring for SWG 107
 - customizing messages for a Reject ending 107
 - Empty access policy action 75, 82
 - local traffic pool, about assigning 31
 - macrocalls about 12
 - per-flow variable 103
 - selecting an SSO configuration 81, 92
 - size limit 12
 - subroutine macrocall about 12
 - unique items for 75, 82
- per-request policy endings
 - about 93
- per-request policy subroutine
 - about 12
 - assigning iRule events 61, 86
- per-request policyaccess policy
 - configuration, using visual policy editor 7
- per-session policy
 - assigning a local traffic pool 31
- preconfigured access policy item
 - adding to an access policy 17
- preconfigured item
 - adding dynamically 17
- predefined session variables
 - updating from Variable Assign 112
- protocol lookup
 - per-request policy item 75

R

- RADIUS Auth
 - and subroutine logon retry 82
- RADIUS class lookup
 - in an access policy 76
- release notes, finding 13
- report
 - per-flow variable 103
- request analytics
 - per-request policy item 79
- Resource allocation
 - session variables 96
- response analytics
 - per-request policy item 78
 - providing web response page for 77
- route domain
 - selecting in an access policy 32

S

- Safe Search
 - enabling 77
- Secure Web Gateway
 - and per-request policy 12
- session management
 - session variables 96
- session variable

- session variable (*continued*)
 - assigning 33
- session variable name
 - \$attr 95
 - \$name 95
 - last 95
- session variables
 - about 95
 - in reports 95
 - logging in an access policy 62, 77
 - using in branch rules 95
 - viewing 95
- sessiondump 101
- SNAT
 - assigning in an access policy 32
 - assigning in network access resource 32
 - assigning in virtual server 32
 - assignment precedence 32
- split domain
 - and session.logon.last.username 23
- SSL bypass setbypassSSL forward proxy traffic
 - bypassing in per-request policy 75
 - per-request policy item 75
 - SSL forward proxy traffic 75
- SSL intercept setinterceptSSL forward proxy traffic
 - intercepting in per-request policy 77
 - per-request policy item 77
 - SSL forward proxy traffic 77
- SSO configuration
 - selecting from a per-request policy 81, 92
- SSO credentials sAMAccountName
 - session variable for 33
 - specifying 33
 - specifying for SSO 33
- start session script variables
 - passing to VMware Horizon View 34
- subsession
 - duration and subroutine properties 12
 - variables 12

T

- Tcl
 - about usage 109
- Tcl example
 - domain\user 111
 - strings, concatenating 111
- Tcl expressions
 - and Variable Assign item 29
- Tcl syntax
 - and iRules 109
 - APM logical operators 109
 - APM rule operators 109
 - brace 110
 - bracket 110
 - expr command 110
 - mcget command 109
- troubleshooting
 - Edge Client for Windows 117
 - verifying trusted server registry keys 117

U

- URL Branching 92
- URL filter lookup
 - per-request policy item 79
- USB redirection
 - disabling for VMware clients 34
 - enabling for VMware clients 34

V

- variable
 - about assigning 33
 - per-flow 103
 - per-flow subroutine 103
- variable assign
 - custom expression 111
 - custom expression example 110
 - domain\user 111
 - strings, concatenating 111
- virtual keyboard
 - about 27
- visual policy editor
 - about 7
 - configuring a per-request policy 7
 - configuring an access policy 7
- visual policy editor branches, about
 - endings, configuring 7
 - macro, adding and configuring 7
 - macrocall, adding 7
 - rectangle, about 7
 - red asterisk, about 7
 - shaded rectangle, about 7
- VMware View
 - client type, detecting 53
- VMware View logon page action
 - about 28
- VMware View Policy
 - about 34

W

- web browser
 - client type, detecting 53
- Windows
 - operating system check 52
- Windows cache and session control
 - and Windows Protected Workspace 46
 - cleaning up after an access session 46
 - in an access policy 46
- Windows file check 46
- Windows Health Agent action
 - dependency on EPSEC software 47
 - in an access policy 47
- Windows Inbox F5 VPN Client
 - client type, detecting 53
- Windows Info
 - session variables 96
- Windows operating system
 - service packs, checking 48
 - updates, checking 48
 - verifying 48

Index

- Windows Process
 - session variables [96](#)
- Windows process action
 - in an access policy [48](#)
- Windows protected workspace
 - and Windows Cache and Session Control [49](#)
 - in an access policy [49](#)
- Windows Registry
 - allowed key, screenshot [115](#)
 - fetching a key [117](#)
 - fetching values from [115](#)
 - GET operator [117](#)
 - in an access policy [50](#)
 - session variables [96](#)
 - specifying trusted servers [116](#)
 - supporting GET operation on client [116](#)
- Windows Registry check
 - 32-bit and 64-bit registry keys [51](#)
 - in an access policy [51](#)
 - supported registry keys [51](#)
- Windows service pack
 - checking client [48](#)