CA Adapter

Cisco IPSec VPN Configuration Guide

r2.2.9

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At http://ca.com/support, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at http://ca.com/docs.

Contents

Chapter 1: Performing Post-Installation Tasks

This document describes the configurations that are required to integrate CA Adapter with the Cisco IPSec VPN appliance to provide secure access to resources.

Note: Before you configure AuthMinder, ensure that Adapter is installed and configured. For information about installing and configuring Adapter, see *CA Adapter Installation and Configuration Guide*.

Authentication Flow Manager (AFM) functions as a proxy between CA AuthMinder and VPN Client. After you install and configure AFM, you must configure AuthMinder to communicate with your VPN server. This section walks you through the process of configuring AuthMinder.

Cisco IPSec VPN uses the **Remote Authentication Dial In User Service** (**RADIUS**) protocol for centralized access, authorization, and accounting management. To enable AuthMinder Server for the RADIUS protocol support, perform the following tasks:

1. Enable the RADIUS protocol in AuthMinder Server.
2. Create a RADIUS client.

For complete information about working with the RADIUS protocol in AuthMinder, see *CA AuthMinder Installation and Deployment Guide*.

Note: CA Adapter still contains the terms Arcot, WebFort, and RiskFort in some of its code objects and other artifacts. Therefore, the CA Adapter documentation also contains occurrences of Arcot, WebFort, and RiskFort. In addition, some of the topics in this guide do not follow the standard formatting guidelines. These inconsistencies will be fixed in a future release.

Checking RADIUS Protocol Support in AuthMinder Server

By default, AuthMinder is enabled to process RADIUS requests. You can use Administration Console to verify the RADIUS protocol status.

Before proceeding with the steps mentioned in this section, ensure that AuthMinder Server *and* Administration Console are successfully installed and configured.

To verify RADIUS protocol support in AuthMinder Server:

1. Log in to Administration Console as a *Master Administrator* by using the following URL:

   http://*App_Server_Host_Name*:*Port_Number*/arcotadmin/masteradminlogin.htm

   In the preceding URL, *App_Server_Host_Name* indicates the host name or the IP address of the application server where you configured Administration Console and *Port_Number* indicates the port at which the server listens to incoming requests.
2. Click the Services and Server Configurations tab.
3. Activate the WebFort tab.
4. Under the Instance Configuration section on the side-bar menu, click the Protocol Management link.

   The Protocol Configuration page opens.
5. In the List of Protocols table, click RADIUS.

   The RADIUS page opens.
6. Ensure that the Protocol Status field is set to Enabled.

   If the protocol is not enabled, then perform the following steps.
7. Select the Change Protocol Status check box.
8. From the Action drop-down list, select Enable**.**
9. Click Save to save your changes.
10. Restart AuthMinder Server.

You have enabled RADIUS protocol support in AuthMinder Server.

Creating RADIUS Clients

You can use Administration Console to create a RADIUS client.

To create a RADIUS client:

1. Log in to Administration Console as a *Global Administrator* by using the following URL:

   http://*App_Server_Host_Name*:*Port_Number*/arcotadmin/adminlogin.htm

   In this URL, *App_Server_Host_Name* indicates the host name or the IP address of the application server where you configured Administration Console. *Port_Number* indicates the port at which the server listens to incoming requests.

2. You can add the RADIUS client at the global level or at an organization level. The page to add the client is the same, but the navigation to the page differs.

   - To add a client at the global level:

     a. Activate the Services and Server Configurations tab.

     b. Activate the WebFort tab.

     c. In the left pane, click RADIUS Client.

   - To add a client at the organization level:

     a. Activate the Organizations tab.

     b. Under the Manage Organizations section, click the Search Organization link to display the Search Organization page.

     c. Enter the complete or partial information of the organization you want to search and click the Search button.

     d. A list of organizations matching the search criteria is displayed.

     e. Under the Organization column, click the *Organization_Name* link for the required organization.

     f. The Organization Information page opens.

     g. Activate the WebFort Configuration tab.

     h. The organization-specific configuration links are displayed in the tasks pane.

     i. In the left pane, click RADIUS Client.

     The RADIUS Configuration page opens.

3. Click Add.

4. The page to add a RADIUS configuration opens. Provide information to add a RADIUS configuration, as described in the following table.

Parameter	Description

Parameter	Description
RADIUS Client IP Address	The IP Address of the RADIUS client using which users authenticate to CA AuthMinder RADIUS Server.
Shared Secret Key	The secret key that is shared between the RADIUS client and CA AuthMinder Server. **Note:** The minimum length of the key is 1 and the maximum length is 512 characters.
Description	A brief description of the RADIUS client. The description helps identify the RADIUS client, when multiple clients are configured.
Authentication Type	Select **RADIUS OTP** as the type of authentication that is used with VPNs.

5. Click **Add** to add the IP address of the new RADIUS client.

   Note: The RADIUS Configuration page also displays the Existing RADIUS Clients table, using which you can update or delete the RADIUS client IP addresses.

   You have created a RADIUS client.

6. For the changes to take effect, refresh the AuthMinder Server instance.

Important! After you configure AuthMinder Server, you must configure the Cisco IPSec VPN appliance to use AuthMinder as an authentication server over RADIUS. See the appropriate Cisco documentation for configuration details.

Chapter 2: Configuring VPN Client

VPN Client is an end-user application that is shipped with Adapter. VPN Client must be installed on the end user's system to enable them to access your VPN server. Users can install VPN Client by using the Arcot-VPN-Client-1.0.2.2-Windows.zip package that contains the client installer file and other configuration files required to connect to your enterprise network.

Before installing VPN Client, you must configure the .properties and .xml files used by the client application. To reduce errors while editing these files, CA strongly recommends that the administrator modifies these files *before* distributing them to the end users. This section covers the following topics:

Note: Ensure that the Cisco VPN Client is installed and configured with a profile to communicate with the VPN Gateway on the system where the VPN Client will be installed and used.

Exploring the VPN Client Package File

The CA-Arcot-VPN-Client-1.0.2.2-Windows-32bit.zip package contains the VPN Client installer (CA-Arcot-VPN-Client-1.0.2.2-Windows-Installer.exe) and the resources directory, which contains the following files and subdirectories:

- The arcotvpnclient.properties file, which contains the configurable parameters that are used by the VPN Client application.
- The properties subdirectory, which contains the following files:
  - arcotvpnclienterrors.properties: Used to customize the error messages displayed by the VPN Client application.
  - avcstrings.properties: Used to customize the element text displayed on the VPN Client user interface.
  - log4j.properties: Used to define the log file location and the level of logging to be done for the VPN Client application.
- The images subdirectory contains the CA logo, which is used while rendering the workflow screens to the end users.
- The screens subdirectory contains XML files, which are used for rendering the VPN Client user interface.
- The add-ons subdirectory contains the CopyFiles.xml file, which is used for specifying the add-on files that need to be copied to the end-user's system.

The following sections describe the contents of these configuration files and explain how to customize these files according to your system requirements.

Note: In the following sections, wherever you need to make a change in the default configuration value, you must first uncomment the respective parameter by removing "**#**" before the parameter entry.

Editing the arcotvpnclient.properties File

The arcotvpnclient.properties file specifies:

- The default and failover URLs to connect to the AFM server, which is also referred to as *Arcot Authentication Proxy* (AAP).
- The mapping between VPN Profile and AAP.
- The network timeout settings to specify the connection timeout and read timeout.
- The network proxy server settings.
- The Cisco VPN client commands that are used to connect, disconnect, and refresh a connection to the Cisco VPN server.
- The connection status messages.
- The location of the image files used in the VPN Client user interface.

The following table describes the various parameters of the arcotvpnclient.properties file that you must configure. The table is divided into parts that correspond to various sections in the properties file.

Important! In addition to the parameters described in the following table, the arcotvpnclient.properties file contains other parameters. However, you should *not* modify them for the VPN integration.

Parameter	Required/ Optional	Description
Cisco VPN Client Profile to AAP URL Mapping This section defines the Cisco VPN server parameters.		

Parameter	Required/ Optional	Description
aap.profile.serverurl. mapping.*Profile_Name*	Optional	If your Cisco VPN client configuration supports multiple VPN server connections, you need to specify them in the aap.profile.serverurl.mapping.*Profile_Name* parameter. The connection details are stored as a VPN Profile at the client end, with each profile representing one VPN server. In the file, uncomment this parameter and specify the mapping. You can create as many profile-to-AAP mapping sections as required. **Note:** If the profile name contains any white space characters, then the profile mapping can be specified by escaping the white space characters with backslash "****". For example, if the profile name is "*profile 1*", the profile mapping entry should be specified as:app.profile.serverurl.mapping.profile\ 1= https://Host_IP:Port_Number/arcotafm/vpn /master_vpn.jsp

Parameter	Required/ Optional	Description
aap.default.serverurl	Required	This AFM URL is used for the profiles that are *not* mapped to any AAP. In the file, uncomment the aap.default.serverurl parameter and specify the AFM URL. By default, the URL is in HTTPS format. If the application server is enabled for HTTPS and if its certificate is not trusted by the JRE used by VPN Client, then include that certificate in the trust store of the JRE. For example, if the application server is using a self-signed certificate. **Note:** You can specify multiple (AAP mapping or URL) using comma as the delimiter, as show in the following example: https://Host:Port_Number/arcotafm/vpn/master_vpn.jsp, https://Host:Port_Number/arcotafm/vpn/master_vpn.jsp
aap.monitoring.jsp	Required	VPN Client sends a request to the JSP file specified in this parameter to check whether AAP is running or not. By default, this parameter is set to arcotauthuiMonitor.jsp file. **Note:** *Do not* change the value of this parameter.
Network Timeouts This section defines the connection timeout and read timeout values between VPN Client and AFM.		

Parameter	Required/ Optional	Description
vpnclient.connection. timeout	Optional	The connection timeout parameteAVCr is used to specify the time interval (in milliseconds) for which VPN Client waits for AFM's response to a new connection request. If VPN Client receives a valid response from AFM within the specified time frame, the connection is establiAVCshed. Else, VPN Client terminates the connection request. By default, this parameter is set to 30000 milliseconds. If you need to specify a different value, uncomment this parameter and specify the required time interval.
vpnclient.read.timeout	Optional	In case of a successful client-server connection, the VPN Client read timeout parameter specifies the time intervaAVCl (in milliseconds) for which VPN Client waits for AFM's response to a request sent by VPN Client. If AFM fails to respond back in the specified time frame, the connection to AFM is dropped. By default, this parameter is set to 30000 milliseconds. If you need to specify a different value, uncomment this parameter and specify the required time interval.
SSL-Related Settings This section defines the protocol used for the SSL communication.		
vpnclient.sslProtocol. Version	Optional	This parameter specifies which version of SSL should be enabled in VPN Client while communicating with the SSL-enabled servers. The possible values are: ■ SSLv3 ■ TLSv1 ■ SSLv2Hello,SSLv3 ■ SSLv2Hello,TLSv1 ■ SSLv2Hello,TLSv1,SSLv3 ■ TLSv1,SSLv3 Default value: SSLv3

Parameter	Required/ Optional	Description
verifyHostName	Optional	This parameter specifies whether the host name in the SSL certificate is verified by the VPN Client application or not. By default, this parameter is set to true, which indicates that the host name in the SSL certificate is verified by the VPN Client application. If this parameter is set to true and there is a mismatch in the host name, then the VPN Client application displays an error message. If you do not want the VPN Client application to verify the host name, set the value of this parameter to false. **Note:** If verifyHostName is set to true, then ensure that the certificate used by AFM matches the domain where it is installed.
Network Proxy Settings The parameters defined in this section are applicable when the end user accesses your enterprise network using a proxy. CA recommends that you *do not* edit these parameters manually, as they are configured through the proxy server settings screen of the VPN Client application. For more information about configuring proxy server settings in VPN Client, refer to the section "Configuring VPN Client to Work With a Proxy Server" in the *CA VPN Client User Guide*.		
proxyHost	Optional	The host name of the proxy server.
proxyPort	Optional	The port number of the proxy server. Default value: 80
proxyAuthenticationRequired	Optional	Determines whether end users are required to authenticate with the proxy server. Possible values are true and false. Default value: true
proxyUser	Required	The username that the end user uses to authenticate with the proxy server.

Parameter	Required/ Optional	Description
proxyPassword	Required	The password that the end user uses to authenticate with the proxy server. **Note:** The proxyUser and proxyPassword parameters cannot be configured through the arcotvpnclient.properties file as they are configured through the VPN Client proxy server settings screen. These parameters should never be commented and should be specified in the Base64 format only.
last.used.proxy.setting	Optional	This parameter specifies the last proxy setting used in VPN Client. Possible values are: ■ direct ■ system ■ manual Default value:direct **Note:** These parameters can also be configured using the VPN Client proxy server settings screen. The parameter configurations made in the arcotvpnclient.properties file can be overridden by changes made subsequently through the VPN Client proxy server settings screen. CA recommends that you configure these parameters by using the VPN Client proxy server settings screen.
Cisco VPN Client Commands This section specifies the Cisco VPN client commands that are used to establish a connection with the VPN server, check the connection status, and disconnect from the VPN server. Additionally, you can specify pre- and post-connection commands, if applicable.		

Parameter	Required/ Optional	Description
vpnclient.connect.command	Required	This parameter specifies the command to establish a connection with the VPN server. For example, the default value of this parameter is as follows: vpnclient.connect.command={base.vpn.client.directory}\\vpngui.exe **-sc** -user "#username#" -pwd "#password#" "#profile#" **Note:** **–** In the preceding command, the {base.vpn.client.directory} variable and the #profile#, #username#, and #password# parameters are replaced by the VPN Client installer, at the time of installation, with the actual values. **–** If you want to display an icon of the Cisco VPN client application in the system tray, then you must set the value of systray.display to false and specify the *connect* command as: vpnclient.connect.command={base.vpn.client.directory}\\vpngui.exe -c -user "#username#" -pwd "#password#" "#profile#" For more information on this feature, see Enabling Automatic Termination of the VPN Client Application (see page 29).
vpnclient.preconnect.command	Optional	This parameter specifies any command that needs to be run before establishing a connection with the VPN server.
vpnclient.postconnect.command	Optional	This parameter specifies any command that needs to be run after a connection with the VPN server has been established.

Parameter	Required/ Optional	Description
vpn.cmd.check.status	Required	This parameter specifies the command to update the connection status. For example, the default value of this parameter is as follows: vpn.cmd.check.status={base.vpn.client.directory}\\vpnclient.exe stat traffic **Note:** In the preceding command, the {base.vpn.client.directory} variable is replaced by the VPN Client installer, at the time of installation, with the actual value.
vpn.cmd.disconnect	Required	This parameter specifies the command to disconnect an active VPN server connection. The default value of this parameter is as follows: vpn.cmd.disconnect={base.vpn.client.directory}\\vpnclient.exe disconnect **Note:** In the preceding command, the {base.vpn.client.directory} variable is replaced by the VPN Client installer, at the time of installation, with the actual value.
profile.directory. location	Required	This parameter specifies the location of the Cisco VPN client profiles on the end-user's system. For example, the default value of this parameter is as follows: profile.directory.location={base.vpn.client.directory}\\Profiles **Note:** In the preceding command, the {base.vpn.client.directory} variable is replaced by the VPN Client installer, at the time of installation, with the actual value.
line.to.grep.for.notconnected	Required	This parameter specifies messages that are returned by the status check command in case of no active connection. For example, the default value of this parameter is as follows: line.to.grep.for.notconnected=No connections exists; Your VPN connection is not active **Note:** You can specify multiple messages by separating them with a semicolon (**;**).

Parameter	Required/ Optional	Description
line.to.grep.for.connected	Required	This parameter specifies messages that are returned by the status check command if an active connection is found. For example, the default value of this parameter is as follows: line.to.grep.for.connected=Time connected
VPN Client Images Location This section specifies the location of image files used in the VPN Client user interface. You can replace these default images by specifying the path of the image files that you intend to use.		
image.appicon	Required	This parameter specifies the location of the image used in the title bar of the VPN Client user interface. By default, it shows the Arcot logo (appicon.gif).
image.connected	Required	This parameter specifies the location of the image file shown in the status bar for an active connection. By default, it is set to display the Arcot logo with green background color (ArcotTrayIconC.gif).
image.disconnected	Required	This parameter specifies the location of the image file shown in the status bar for an inactive connection. By default, it is set to display the Arcot logo with red background color (ArcotTrayIconDC.gif).

Editing the arcotvpnclienterrors.properties File

The arcotvpnclienterrors.properties file specifies the error messages that are displayed to the end user while working with the VPN Client application. A snippet of the arcotvpnclienterrors.properties file is as follows:

```
##### Client side error messages #####
# This file contains error messages that are used by screens on the client system.
...
# Error messages for client configuration errors.
profilenotfound=No existing vpn profile found.
...
# Error messages for server communication failures.
allaapserversdown=The proxy server or the VPN server is not available. Please try after
some time.
communicationfailure=Communication failure, please try again.
...
# Error messages for proxy server issues.
proxyserverauthfailure=The username or password you entered for the proxy server
authentication is invalid. Please try again.
proxydetailsincomplete=Username field is required.
...
# Error message on failure to launch browser
browserFailed=The browser failed to launch.

# Error message when communication with current server fails in
# middle of a flow, and the client restarts the flow
failovererror=Communication with server failed. Please retry.

# Error message when tried to launch application when one instance is running
applicationalreadyrunning=Another instance of application is already running. Exit
already running instance to relaunch.

# Error message when underlying VPN already connected
underlyingvpnalreadyconnected=You are already connected to the VPN network!

# Error while parsing error xml
error.parsing.errorxml= Internal error in parsing XML. Please contact administrator.

# Error message on failure to receive OTT with in time
error.ott.timeout=Arcot VPN Client was not able to get your credentials from A-OK in
time. Please try again.

# Messages for System Proxy setting problems.
systemsettingsunavailable=Error occurred while detecting system settings. Direct
connection will be used.
...
```

The arcotvpnclienterrors.properties file uses the following pattern:

```
<ParameterName>=<Value>
```

You can customize an error message by modifying the Value field of the respective parameter. However, you *must not change* the ParameterName.

Editing the avcstrings.properties File

The avcstrings.properties file is used to customize the following elements of the VPN Client user interface:

- The version and copyright information in the footer.

Note: If you do not want to display the version and copyright information in the footer, you can turn these parameters off by either commenting the footer parameters (window.footer.version and window.footer.copyright) or leaving them blank.

- The application title.
- The field names displayed in the proxy server settings screen.
- The action button names and their shortcut keys.
- The label for the drop-down list for selecting a profile.
- The main menu and pop-up menu labels.
- The system tray messages.

A snippet of the avcstrings.properties file is as follows:

```
##### Client side screen messages #####
# This file contains messages that are used by screens on the client system.
# If you want to build up locale-specific support, you need to create
# avcstrings_<language>_<country>.properties inside this folder.
########################################

# Version and copyright information
window.footer.version=Version 1.0.2.2
window.footer.copyright=Copyright 2012, Arcot Systems, Inc.

# Window title
application.title=Arcot VPN Client

# Title displayed inside client body
application.labels.title=Arcot VPN Client

# Proxy Authentication
proxyauth.form.instruction=Proxy server requires authentication
proxyauth.form.USERNAME=Username:
proxyauth.form.PASSWORD=Password:
proxyauth.form.REMEMBER=Remember credentials

# Proxy Settings Field names
proxysettings.form.USERNAME=Username:
proxysettings.form.PASSWORD=Password:
proxysettings.radio.PROXY_ENABLE=Proxy Required:
proxysettings.radio.PROXY_DISABLE=Direct Connection (No proxy)
proxysettings.radio.SYSTEMSETTINGS_ENABLE=Use System Settings (Internet Options)
```

```
proxysettings.checkbox.AUTHENTICATE=Authentication Required
proxysettings.form.PROXYHOST=Host
proxysettings.form.PROXYPORT=Port
proxysettings.checkbox.AUTODETECT_PROXY=Auto Detect Proxy

# Proxy setting window instruction
proxy.settings.instruction=Please Configure proxy details.

# buttons used in screens together with hokeys (shortcuts) to invoke them
application.buttons.OK=OK
application.buttons.OKHOTKEY=O
application.buttons.SUBMIT=Submit
application.buttons.SUBMITHOTKEY=S
application.buttons.CANCEL=Cancel
application.buttons.CANCELHOTKEY=C

# select profile dropdown box in case of profile url mappings
select.profile.label=Connection Entry:

# Menu names.
tools.menu=Tools
tools.menuitem.enableproxysettings=Enable Proxy
tools.menuitem.changeproxysettings=Proxy Settings

# TextField Menu Items.
textField.Copy=Copy
textField.Cut=Cut
textField.Paste=Paste
textField.Select=Select All

# system tray messages
systray.connected=Connected to VPN.
systray.disconnected=Disconnected from VPN.
```

Editing the CopyFiles.xml File

The VPN Client installer can copy extra files that might be required by VPN Client or the Cisco VPN client application to work successfully. These files could include the profile configuration or other system files that might be required for successfully running the client application.

The CopyFiles.xml file is used to specify the location where these extra files need to be copied on the end-user's system. To distribute these extra files with the VPN Client installer, perform the following tasks:

1. Copy the files that need to be distributed with the installer to the \resources\add-ons directory.
2. In the CopyFiles.xml file, specify the location on the end-user's system where these files need to be copied.

   A sample CopyFiles.xml file is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
- <B>How it works</B>
-<B> Install</B>
-Document element
-   <B>OS</B>
-    To perform OS Specific install actions
-<B>PostInstall</B>
-All the children of this tag should be executed in PostInstall phase only.
-Each of the above can contain either of these only.
- <I>COPY</I>,
-<B>1. COPY</B>
-<I>from_location</I> location from where to copy, This need to be either
-absolute location or some IA variable.
-
-<I>to_location</I>  is the location where the copying will done. This need to
be either
-absolute location or some IA variable.
-
-
-<I>type</I> is  to denote if the command to be applied is on directory.
-
-
-
-->

<!DOCTYPE Install [
  <!ELEMENT Install(OS+)>
  <!ELEMENT OS (PostInstall)>

  <!ELEMENT PostInstall(COPY)+>
  <!ELEMENT Object(#PCDATA)>
```

```
  <!ELEMENT COPY(Object+)>

  <!ATTLIST OS name CDATA #REQUIRED>
  <!ATTLIST COPYfrom_location CDATA #REQUIRED>
  <!ATTLIST COPYto_location CDATA #REQUIRED>
  <!ATTLIST COPYtype CDATA #REQUIRED>

  <!ATTLIST Objectname CDATA #REQUIRED>


]>


<Install>
<OS name="windows">
<PostInstall>
<!--
USER_INSTALL_DIR : Location of Arcot VPN Client installation (ARCOT HOME
directory)
VPN_CLIENT_INSTALL_LOCATION : Location of VPN client (e.g. Cisco VPNClient)
installation (already installed)
EXTRACTOR_DIR : Directory containing Arcot-VPN-Client-*.*-Windows-Installer.exe
-->

<!-- Example for copying a directory -->
<!--
<COPY to_location="$VPN_CLIENT_INSTALL_LOCATION$+$/$"
from_location="$EXTRACTOR_DIR$+$/$+resources+$/$+add-ons+$/$"
type="directory">
<Object name="Directory Name"/>
</COPY>
-->

<!-- Example for copying a file -->
<!--
<COPY to_location="$VPN_CLIENT_INSTALL_LOCATION$+$/$"
from_location="$EXTRACTOR_DIR$+$/$+resources+$/$+add-ons+$/$" type="file">
<Object name="File Name"/>
</COPY>
-->
</PostInstall>
</OS>
</Install>
```

The following table describes the XML tags and the variable names that are used in the CopyFiles.xml file.

Element	Description
to_location	Specifies the path where the add-on files need to be copied on the end-user's system.
$VPN_CLIENT_INSTALL_LOCATION$+$/$	Contains the location of the VPN client installed on the end-user's system. Typically, this variable should not be changed. However, if you want to store the add-on files on a location other than the VPN client installer, then replace this variable with the required location. For example, C:\Program Files\Profiles.
$USER_INSTALL_DIR$+$/$	Contains the install location of the VPN Client on the end-user's system.
from_location	Specifies the path from where the add-on files need to be picked up for copying. By default, this is set to the \resources\add-ons directory that is extracted from the VPN Client zip file.
$EXTRACTOR_DIR$+$/$+resources+$/$+add-ons+$/$	Contains the location of the add-ons directory that is extracted from the VPN Client zip file.
type	Specifies the type of object being copied. It can be assigned one of the following values: ■ **directory**: the object that needs to be copied is a directory. ■ **file**: the object that needs to be copied is a file.
Object name	Specifies the name of the object (directory name or file name) that needs to be copied.

Note: You can now either customize the VPN Client interface, as discussed in "Customizing the VPN Client Interface" (see page 29), or directly proceed to "Creating the Final Package" (see page 37).

Chapter 3: Customizing the VPN Client Interface

This section discusses how to customize the VPN Client application to enable or disable some of its features. It includes information on performing the following customizations:

Note: The configurations explained in this section are optional.

Enabling Automatic Termination of the VPN Client Application

By default, VPN Client is configured to run in the background and the application icon appears in the system tray. In this case, the Cisco VPN client application icon does not appear in the system tray. To change this behavior, you need to make the following changes in the arcotvpnclient.properties file:

1. Set the value of systray.display parameter to False.
2. Set the value of vpnclient.connect.command parameter, as shown below:

```
{base.vpn.client.directory}\\vpngui.exe -c -user "#username#" -pwd "#password#"
"#profile#".
```

Making these changes in the arcotvpnclient.properties file configures the VPN Client application to automatically terminate once the user is successfully authenticated by AFM, and the control passes to the Cisco VPN Client application. In addition, the Cisco VPN Client application icon appears in the system tray.

Enabling Auto Copy for Selected Text to Clipboard

To enable the auto copy feature for a text field on the VPN Client user interface, you need to add autoCopy in the field declaration. The following example shows how you can add this attribute in a text field declaration:

```
<textField name="<%= strProps.getString("aidauth.form.USERID") %>" colspan="2"
value="<%= StringEscapeUtils.escapeXml(sd.getLoginID()) %>" focus="<%=
!loginidpresent %>" autoCopy="true" />
```

Customizing Splash Screen

The Splash screen is stored as vpnsplashscreen.png in the resources\images folder of the VPN Client package. You can replace the Splash screen by overwriting the vpnsplashscreen.png image file with a new image of your choice. You need to make sure that the new image file is stored with the same name, vpnsplashscreen, and is in the .png format.

Enabling Default Button for the VPN Client User Interface

You can assign a default action button for every screen of the VPN Client application. This functionality can be achieved by editing the XML or JSP file used to render the user interface and setting the default parameter to true for the desired button. As a result, when a user presses the **Enter** key on that screen, the action defined for the default button gets executed. The following example shows how you can add this parameter to a button named **Login**:

```
<button name="Login" actionName="submitPage" padding-left="10" padding-right="10"
hotkey="L" default="true"/>
```

Disabling Fields on the VPN Client User Interface

By default, when a user submits a login form, various controls on the login form are disabled. This behavior can be customized by adding the following attributes in the respective XML or JSP file:

- **disableFormOnSubmit**: This parameter disables all form controls, such as text fields or drop-down lists, present on a form. You must specify this parameter in the actionName attribute of a button control. When a user clicks a button with actionName attribute set to disableFormOnSubmit, then all fields present on that form are disabled.

  The following code snippet shows the usage of the disableFormOnSubmit attribute.

  ```
  <button name="<%= strProps.getString("application.button.LOGIN") %>"
  actionName="disableFormOnSubmit,signChallenge,rememberUsername,storeRemember
  Fields,submitPage" colspan="1" padding-left="10" padding-right="20" hotkey="<%=
  strProps.getString("application.button.LOGINHOTKEY") %>" changedName="<%=
  strProps.getString("aidauth.button.changedname.LOGIN") %>" default="true"/>
  ```

- **disableOnSubmit**: This parameter is used to specify whether a control would get disabled or not if the disableFormOnSubmit parameter is specified for a button. By default, all fields on a form are set to be disabled when the button with the disableFormOnSubmit parameter is clicked. However, you can change this behavior by setting disableOnSubmit attribute to false for any form field that you do not want to disable when the disableFormOnSubmit method runs.

  The following code snippet shows the usage of the disableOnSubmit attribute.

  ```
  <button name="<%= strProps.getString("application.button.CANCEL") %>"
  actionName="closeApplication" colspan="1" padding-left="0" padding-right="10"
  hotkey="<%=strProps.getString("application.button.CANCELHOTKEY") %>"
  disableOnSubmit="false"/>
  ```

- **changedName**: This parameter is used to specify a new label or text for a form button after a user has clicked the button. For example, if you want the text on the **Login** button to change to **Connecting** after a user has clicked the **Login** button, you can use the changedName attribute to achieve this.

  The following code snippet shows the usage of the changedName attribute.

  ```
  <button name="<%= strProps.getString("application.button.LOGIN") %>"
  actionName="disableFormOnSubmit,signChallenge,rememberUsername,storeRememberF
  ields,submitPage" colspan="1" padding-left="10" padding-right="20" hotkey="<%=
  strProps.getString("application.button.LOGINHOTKEY") %>" changedName="<%=
  strProps.getString("aidauth.button.changedname.LOGIN") %>" default="true"/>
  ```

Enabling Multiple Organization Support

VPN Client and AFM can be customized to authenticate users from different AuthMinder organizations that are configured in AuthMinder Server. The users are shown a list of organizations on the login screen of the VPN Client application. If multiple organizations are defined in the aidauth.jsp file, then a drop-down list appears with the list of organizations. This list can be customized by editing the following line in the aidauth.jsp file.

```
String orgnamelist = "DEFAULTORG,GROUPA,GROUPB"
```

Note: On Apache Tomcat, the aidauth.jsp file is available in the <*Application_Server_Home*>/webapps/arcotafm/vpn directory. If you are using other application servers, then refer to the application server vendor documentation for the corresponding path.

In the above example, organization names are separated by a comma (**,**), which is also the default delimiter. To change the default delimiter, you must make the following changes:

1. Add a new parameter named orglistDelimiter in the arcotvpnclient.properties file, and specify the new delimiter. In the following example, the delimiter is changed to a semicolon:

   orglistDelimiter=;

2. Save the arcotvpnclient.properties file.

Enabling Automatic Saving and Retrieval of User Information on the Client Side

To configure the VPN Client application to automatically save and load user information on the client side, you must set the value of the remember parameter of a form field to true in the aidauth.jsp file. Configuring the remember parameter allows the field value to be stored in and retrieved from the local file system.

To auto-load user information, in addition to setting the remember parameter, you must include the loadRememberFields parameter in the onLoad attribute of a document tag. This means that as soon as the UI is rendered, the fields configured with the remember=true parameter are populated with the locally stored values.

Similarly, to store the values entered in the fields with remember parameter, call the client-side method, storeRememberFields. This method, which reads and stores values of the configured fields in the local file system, must be called when the form's **Submit** button is clicked.

Additionally, to store the values entered in the fields with remember parameter, you must include the storeRememberFields parameter in the actionName attribute of a button. When you click this button, the values are stored in the fields with the remember parameter.

The following example shows how to configure the information entered in the user ID and profile fields so that they can be stored and retrieved automatically:

```
<document onLoad="populateProfiles,loadRememberFields" width="300" height="350"
xmlns="http://integrations.arcot.com/vpn/client/XMLSchema/1.0">
<body bgcolor="255,255,0">

<form actionUrl="controller_vpn.jsp">
<textField name="userid" colspan="2"focus="true" remember="true"/>
<select name="profile" colspan="2" remember="true"><font name="Arial"
size="12"/></select>
<button name="Login" actionName="storeRememberFields,signChallenge,submitPage"
padding-left="10" padding-right="20" hotkey="L" default="true"/>
<button name="Cancel" actionName="closeApplication" colspan="1" padding-left="0"
padding-right="10" hotkey="C"/>

</form></body></document>
```

Customizing the Height and Width of the VPN Client User Interface

You can customize the height and width of the VPN Client user interface by changing the height and width attributes in the document tag of the XML or JSP file used to render the user interface. The following example shows how you can customize the height and width of the VPN Client user interface:

```
<document onLoad="populateProfiles,loadRememberFields" width="300" height="350"
xmlns="http://integrations.arcot.com/vpn/client/XMLSchema/1.0">
```

Invoking the Default Web Browser Through VPN Client

The VPN Client application can be configured to open a specific Web page in the default Web browser of the user's system. To add this functionality, you need to add the following code in the XML or JSP file used to render the user interface:

```
<textField hidden="true" name="targetURI" value="http://www.arcot.com"/>

<button name="<%= strProps.getString("application.button.BROWSE") %>"
actionName="invokeBrowser" disableOnClick="false" hotkey="<%=
strProps.getString("application.button.BROWSEHOTKEY") %>"/>
```

The first line of code adds a hidden text field that stores the URL of the Web page that opens in the Web browser. In the example, the value attribute of the text field is set to *http://www.arcot.com*. You can change this attribute to a desired value.

The second line of code adds a button named Browse on the user interface. When a user clicks the Browse button, the default browser is invoked and the Arcot home page is displayed.

Customizing the Time Interval Between Connection Status Checks

By default, VPN Client checks the connection status through the Cisco VPN client application every five seconds. You can change the default time interval by including the vpn.status.check.snooze.time parameter in the Cisco VPN Client Commands section of the arcotvpnclient.properties file and setting it to a desired value. Only positive integers are acceptable as valid values and this command works only when VPN Client is running in the *system tray*.

In the following example, the time interval to check the connection status is set to 1 second:

```
vpn.status.check.snooze.time=1
```

Chapter 4: Creating the Final Package

After you have performed the configurations that are discussed in the topics "Configuring VPN Client" (see page 11) and "Customizing the VPN Client Interface" (see page 29), you must create the final VPN Client package file for your end users to include the configuration changes.

Perform the following steps to create final package:

1. Replace the default .properties and .xml files in the resources directory with the newer files that you created by using the procedures described earlier in this section.
2. Create an installer.properties file with the following data:

   VPN_CLIENT_INSTALL_TYPE=CISCO

   Ensure that the Arcot-VPN-Client-1.0.2.2-Windows-Installer.exe and installer.properties files are available in the same location.
3. Re-create the final installation package file along with the installer.properties file.