Abstract
This white paper explains how to diagnose and troubleshoot Netegrity SiteMinder SSO issues in the EMC® Documentum® Webtop application.

March 2011
Table of Contents

Executive summary .................................................................................................................. 4
Audience ................................................................................................................................. 4

What is Netegrity SiteMinder authentication? ................................................................. 4
How does Netegrity SiteMinder work? .............................................................................. 4
How does Netegrity SiteMinder authentication work with the Documentum Webtop application? .................................................................................................................. 5

Deployment topology .......................................................................................................... 6

Troubleshooting SSO issues ............................................................................................... 7
Settings in the Webtop application ......................................................................................... 7
Settings in Web agent.conf ..................................................................................................... 8
Settings on Content Server .................................................................................................... 8
Logs to track on the client machine ....................................................................................... 10
Logs to track on the application server machine ................................................................. 10
Logs to track on the Content Server machine ................................................................. 10
Logs to track on the Policy Server machine ........................................................................ 10
Logging information for the Web Agent .............................................................................. 11
Recording messages in a log file for the Web Agent .......................................................... 11
Setting the log level for the Web Agent ............................................................................. 11
Web Agent log file rollover ................................................................................................. 12
Displaying Web Agent log messages in a console ............................................................... 13
Troubleshooting Web Agent and Web server issues .......................................................... 13
Interpreting Web Agent error codes ..................................................................................... 15
Troubleshooting Policy Server settings ............................................................................... 19
Known issues ......................................................................................................................... 22

Conclusion ............................................................................................................................ 24
Executive summary

This white paper explains how to diagnose and troubleshoot Netegrity SiteMinder single sign-on (SSO) issues in the EMC® Documentum® Webtop application.

This white paper specifies installation files and configuration steps for Netegrity Policy Server and Web Agent, a Web server, an application server, Content Server, and the Webtop application.

This white paper also explains the settings that must be set on the Policy Server to troubleshoot Netegrity Web Agent and Web server issues. It also lists the known issues encountered by Webtop when integrated with the Netegrity SiteMinder SSO application.

Audience

This white paper is intended for those who need to troubleshoot or diagnose issues that occur in Webtop, which is integrated with Netegrity SiteMinder SSO functionality.

What is Netegrity SiteMinder authentication?

Netegrity SiteMinder provides an SSO solution for enterprises that have multiple intranet Web servers requiring authentication. The Netegrity SiteMinder solution relies on agents and a central Policy Server to provide seamless authentication. User credentials are generally available in the SMSESSION cookie that is set by the agent on the client machine.

How does Netegrity SiteMinder work?

When a client sends a request for a protected resource from the Web server where a SiteMinder Web Agent is configured, the SiteMinder Web Agent verifies the request and validates the credentials provided by the user. Then the Web Agent delegates the request to the Netegrity Policy Server, which displays the Login screen to the user based on the authentication scheme configured in the Realm tab of the SiteMinder Domain dialog box. When the user provides the credentials in the Login screen, the Netegrity Policy Server authenticates the user and sends the credential cookie to the Web Agent on the Web server to verify the cookie. The Policy Server directly verifies this cookie for subsequent requests. Figure 1 illustrates how Netegrity SiteMinder authentication works.
How does Netegrity SiteMinder authentication work with the Documentum Webtop application?

A connector is required to ensure requests to application servers go through the HTTP server. As a result, the Web Agent captures requests to the application server (which are requests to the HTTP server). This is done completely independent of the Web Agent configurations defined in the Netegrity SiteMinder Policy Server and enables the Web Agent to understand that the application is working successfully on the HTTP server.

A session cookie is set on the client machine in a cookie called “SMSESSION” that contains the SiteMinder user credentials. This enables the Webtop application to pick up the username from the SM-USER part of the HTTP header, and the ticket returned from authentication in the SMSESSION part of the header, and send them to the Content Server plug-in. The SM-USER part and SMSESSION part are put into the send headers by the Web Agent after user authentication. The repository must be set up to load the dm_netegrity plug-in.
All URL-related operations that go through the Web server are subject to Web Agent interception. If the URL is applicable to a realm, and the header contains username and ticket details, the Web Agent uses the header for authorizing the user. Alternatively, the user is prompted to provide credentials for authentication.

The Content Server Netegrity plug-in seeks authorization using the username and an SSO token. The token allows the Policy Server to determine all parameters associated with the token that is issued, such as the path requested that generated the token, which allows the Policy Server to determine the realm to which the token applies. The following figures illustrate the deployment topology of a Webtop and Netegrity SiteMinder SSO environment, and how SiteMinder authentication works in the Webtop application.

**Deployment topology**

![Deployment topology diagram](image)

*Figure 2. Deployment topology in the Documentum Webtop environment*
Troubleshooting SSO issues

Settings in the Webtop application

The SSO scheme is configured in the app.xml file. This file contains the HTTP header name and cookie name that the scheme will search for to obtain the credentials to pass on to the Trusted Content Service.

The following extract shows the location of the <ticket_cookie> and <user_header> tags and their contents:

```
<application>

...  

  <authentication>

  <!-- Default domain and docbase to authenticate against -->

  <domain></domain>
  <docbase>devprogDocbase</docbase>

  <!-- Class that provide the authentication service -->

  <service_class>com.documentum.web.formext.session.AuthenticationService<
  /service_class>

    <!-- Single Sign-On authentication scheme configuration -->

    <sso_config>
    <ecs_plug_in>dm_netegrity</ecs_plug_in>
    <ticket_cookie>SMSESSION</ticket_cookie>
    <user_header>SM-USER</user_header>
    </sso_config>

  </authentication>

  ...

</application>
```

The “Trusted Content Service” password must have the following form:

DM_PLUG-IN=[Plug-inName]/[Ticket or Password]

where,

[Plug-inName] is the name of the server-side plug-in used to authenticate the user.

[Ticket or Password] is the ticket or password of the user.

The class com.documentum.web.formext.session.SSOAuthenticationScheme assumes that the ticket is stored as a cookie in the HTTP request, and the username is stored as a HTTP header. The name of the cookie that contains the ticket and the
The SSOAuthenticationScheme class's authenticate method reads the cookie and header values, creates the password in the specified format, and authenticates the user. The default repository against which the user is authenticated is specified in the <docbase> element in the app.xml file. If the authentication against the specified repository fails, the login component is invoked, and the Web Development Kit calls the getLoginComponent() in the SSOAuthenticationScheme class, which returns the value “sso_login”. WDK then launches the getLoginComponent() component. This component shows the list of repositories so the user can select an alternative repository. The username and password credentials are also read from the HTTP request.

**Settings in Web agent.conf**

When the Web Agent is installed, the Web Agent Configuration Wizard generates a WebAgent.conf file for each Web server configured for a new Agent.

The new WebAgent.conf file contains the following default parameters required to perform the basic Web Agent operations. Ensure that the default settings are correctly configured:

- **AgentConfigObject**—Contains the name of the configuration object of the Web Agent. If you change this name, then the Web Agent will use another Agent Configuration Object.

- **HostConfigFile**—Contains the path to SmHost.conf, a file that results from a successful registration of the host computer as a trusted host. All agents on the computer share this file, by default, and the trusted host registration must occur before an agent can operate.

- **EnableWebAgent**—Ensures that the agent protects the Web server’s content. Ensure that this parameter is set to **Yes** when the required Policy Server rules and Web Agent configuration settings are defined.

**Settings on Content Server**

Documentum provides a Web-based SSO solution using the Netegrity SiteMinder authentication plug-in. You must ensure that Content Server is configured to use the dm_netegrity_auth plug-in.

Authentication plug-ins provide an alternate way to perform user authentication. Documentum provides the Netegrity SiteMinder (dm_netegrity_auth) authentication plug-in with Content Server. This plug-in enables you to use the Netegrity SiteMinder Policy Server with Content Server by authenticating the Netegrity SiteMinder token against the SiteMinder Policy Server.

The plug-in takes initialization parameters from an initialization file located in the same directory as the module. The file contains all the parameters required to connect to the SiteMinder Server, such as hostname and ports. Ensure that the...
original dm_netegrity_auth.ini initialization file located in
%DM_HOME%\install\external_apps\authplug-ins\netegrity is copied to the
%DOCUMENTUM%\dba\auth directory.

Ensure that the $DOCUMENTUM/dba/auth/dm_netegrity_auth.ini file contains all
mandatory parameters set to connect the SiteMinder Server.

The contents of the dm_netegrity_auth.ini file and some standard settings are as
follows:

#This is a sample configuration file for the Documentum-Netegrity plug-
in
[DM_NETEGRITY_AUTH_CONF]
#name of the agent protecting the Documentum web application
#this is a mandatory parameter
#e.g.: agent_name = Webtop-agent
agent_name = Webtop_agent

#shared secret between the Netegrity Web Agent and Policy Server
#this is a mandatory parameter
#e.g.: shared_secret =
shared_secret = password

#IP address of Netegrity Policy Server. For failover and load
balancing,
#you can specify more than one, in which case they must be
separated by
#commas. No spaces are allowed.
#this is a mandatory parameter
#e.g.: policy_server_ip = 173.134.21.239,173.134.21.239
policy_server_ip = 10.10.10.10
#Netegrity Policy Server accounting port
#Defaults to 44441
#accounting_port =
#Netegrity Policy Server authentication port
#Defaults to 44442
#authentication_port =
#Netegrity Policy Server authorization port
#Defaults to 44443
#authorization_port =
#Failover mode. This is used if more than one Policy Server
#is specified in policy_server_ip. Supported values are 0 and 1.
#Defaults to 0. Refer to Netegrity documentation for details.

#failover_mode =

@Resource - The Resource to be protected.
resource=/Webtop

Logs to track on the client machine

Collect the following logs on the client machine to troubleshoot SSO-related issues:

- **Charles Log**—Captures the Charles trace that enables the engineer to view all HTTP and SSL/HTTPS traffic between the client machine and the server. This includes requests, responses, and the HTTP headers (which contain the cookies and caching information). Use this log to view request/response, request/response headers, cookies, JSESSIONID, and the SMSESSION cookie (SiteMinder cookie).

- **Java Console Log**—Captures the Java console log that enables the engineer to view applet-related issues, if cookies are in place, and browser JDK/JRE-related information.

Logs to track on the application server machine

Collect the following logs on the application server machine to troubleshoot SSO-related issues:

- Access log
- Error log on the Web server
- Application server log
- Web Agent log
- WDK log with the following settings enabled in TraceProp.properties:
  
  com.documentum.web.common.Trace.SESSIONENABLEDBYDEFAULT=true
  com.documentum.web.formext.Trace.SESSION=true

Logs to track on the Content Server machine

Collect the following logs on the Content Server machine to troubleshoot SSO-related issues:

- Repository log with the Authentication trace enabled
- dm_netegrity log

Logs to track on the Policy Server machine

Collect the following logs on the Policy Server machine to troubleshoot SSO-related issues:

- smps.log
Troubleshooting SSO Authentication with Netegrity SiteMinder in EMC Documentum Webtop

Logging information for the Web Agent

Use the logging function to monitor the performance of the Web Agent and its communications with the Policy Server. You can configure the Web Agent to log messages in the command prompt window, and to write messages to a log file. You can also specify the types of messages to log. You can set up logging in the WebAgent.conf file besides other agent parameters before central configuration is performed. However, when you centrally configure the Web Agent, and set values for the logging parameters in the Policy Server Administrator Console, the new values will override the values that you set locally in the WebAgent.conf file, unless you set the value of the AllowLocalConfig parameter to Yes.

Recording messages in a log file for the Web Agent

To write messages to a log file:

1. Set the LogFile parameter to Yes.
2. Specify a filename in the LogFileName parameter, such as:
   
   ```
   LogFile="Yes"
   LogFileName="/opt/server/admin/logs"
   ```

   Enable the LogAppend parameter to add logging information to an existing log file instead of rewriting the entire file every time logging is invoked. An example is:

   ```
   LogAppend="yes"
   ```

Setting the log level for the Web Agent

You can configure the Web Agent to generate different levels of log messages and then display them in a console window or a file. Choosing a log level facilitates troubleshooting because log levels determine the severity and extent of the logged messages. This allows you to control the detail that the Web Agent includes in a log.

To change the log level, set the LogLevel parameter to 0, 1, or 2. The following table lists the log levels you can select.

- `smaccess.log`

Note: The logs are located in the path C:\Program Files\netegrity\SiteMinder\log.
Table 1. Log levels

<table>
<thead>
<tr>
<th>Log Level</th>
<th>Type of Messages</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Critical error messages only with the least amount of detail. For example, the Web Agent not being able to communicate with Policy Server. This is the default log level.</td>
</tr>
<tr>
<td>1</td>
<td>Warning error messages, trace messages, and flow state messages. These messages provide information about processes without details.</td>
</tr>
<tr>
<td>2</td>
<td>Data messages of the least importance but with the highest level of detail, such as header details and cookie variables.</td>
</tr>
</tbody>
</table>

For each log level you select, the Web Agent prints messages for that level and messages from any lower level. For example, if you choose level 0, the Web Agent prints only those messages. If you choose level 2, the Web Agent prints all messages from levels 0 through 2.

Initially, you must leave the log level at 0 so the Web Agent logs only critical errors. If you want to audit your site’s activity more closely, change the log level to 1 or 2. The log level is changed dynamically. Do not stop and restart the Web server. Within approximately 30 seconds (the agent’s default Policy Server polling interval) of restarting the Web server, you will see new messages in the console window or in the log file.

Web Agent log file rollover

When data is appended to a log file continually, the file can grow to an unmanageable size. Rolling log files enables you to schedule a log file rollover based on a time interval and a file size limit.

Three parameters in the WebAgent.conf control log file rollover are as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td>RollingLog=[&quot;YES&quot;</td>
<td>&quot;NO&quot;]</td>
</tr>
<tr>
<td>RollingLogPeriod =&lt;#&gt;</td>
<td>Sets the number of hours between rollovers to #. When this value is 0, rollover is disabled.</td>
</tr>
<tr>
<td>RollingLogSizeLimit=&lt;#&gt;</td>
<td>Sets the maximum log file size to # KB.</td>
</tr>
</tbody>
</table>

If you set the RollingLogPeriod and RollingLogSizeLimit parameters, log file rollovers and log file sizing will be effective. Rollover always occurs after the specified "log
periods." It also occurs when the size limit is reached; for example, if RollingLogPeriod=4 and RollingLogSizeLimit=100:

- If the log file size passes 100 KB at 3:12, rollover will occur at 3:12.
  
  `myfile.log.09-23-2002-03-12-07`

- Another rollover will occur at 4:00, regardless of the log file size.
  
  `myfile.log.09-23-2002-04-00-00`

- Subsequent rollovers will occur on the 4-hour targets and whenever the file size reaches 100 KB.

**Displaying Web Agent log messages in a console**

You can display log messages in a command prompt window by configuring the LogConsole parameter.

To configure the LogConsole parameter:

1. In the WebAgent.conf file, set the LogConsole parameter to **Yes**.
2. In the Services control panel, restart the Web server.

To stop displaying messages in the Console window, set the LogConsole parameter to **No**.

**Troubleshooting Web Agent and Web server issues**

- **Issue:** The Web server does not prompt for a username and password.

  **Steps to troubleshoot:**
  
  a. Check if the resource is defined correctly in the Policy Server.
  b. Make sure the Web Agent is enabled.
  c. Make sure the realm and rule are defined properly.

- **Issue:** The Web server always prompts for a username and password, but authentication fails.

  **Steps to troubleshoot:**
  
  a. Check if the resource is defined correctly in the Policy Server.
  b. Make sure that a user has been bound to a policy.
  c. Check that a rule is defined for the policy.
• **Issue:** Apache Web server will not start or restart when the Web Agent is enabled.

**Steps to troubleshoot:**

**If the Web server fails to start:**

Check the Apache error log located in this path:

<apache_server_location>/logs/error.log and look for SiteMinder errors.

If you start the Apache Web server from multiple user accounts, you may have orphaned semaphores on your system:

Reboot or use the *ipcrm -s* command to remove the orphaned semaphores. It is recommended that you always start the Web server from the same user account.

**If the Web server fails to restart:**

Do not use the *restart* command. Use the *stop* and *start* commands to restart the server.

• **Issue:** The Web Agent loads successfully but fails to communicate with the Policy Server.

**Steps to troubleshoot:**

a. Ensure that the WebAgent.conf file includes the line `EnableWebAgent="yes"`. The default setting is “No”.

b. Ensure that the Web Agent name and the shared secret entry match those specified in the Policy Server. The shared secret entry is case-sensitive.

c. Verify whether the Web Agent has TCP connectivity to the Policy Server. If a firewall exists between the Web Agent and the Policy Server, ensure that TCP ports 44441, 44442, and 44443 are not blocked by the firewall for two-way traffic.

• **Issue:** Although you reinstall the Web Agent, the WriteLine Failed error occurs and the hostname is not logged in the Webagent.conf file, error -1.

**Steps to troubleshoot:**

Make sure you remove the WebAgent.conf file from the Web server sub-directory after uninstalling the Web Agent.

• **Issue:** The extensions configured for the Web Agent to ignore do not work. The Web Agent continues to send an authorization request to SiteMinder.

**Steps to troubleshoot:**

Verify the value set for the *IgnoreExt* parameter. The string must be comma-separated and must not contain spaces. If there is more than one section in the URL that contains the period (.), the Web Agent will always send an authorization request.
**Interpreting Web Agent error codes**

When the Web Agent encounters problems, it generates error codes, which help you diagnose problems with SiteMinder operations. For example, if the Web Agent is unable to reach the SiteMinder authentication server, the Web Agent displays error code 20-0002.

The following tables list all Web Agent error codes, their meanings, and action to take.

**Table 2. HTTP header parsing error codes**

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Meaning</th>
<th>Action to Take</th>
</tr>
</thead>
<tbody>
<tr>
<td>10-0001</td>
<td>Unable to read 'SERVER_NAME' HTTP variable.</td>
<td>Check that the Web browser and Web server are HTTP 1.0-compliant.</td>
</tr>
<tr>
<td>10-0002</td>
<td>Unable to read 'URL' HTTP variable.</td>
<td></td>
</tr>
<tr>
<td>10-0003</td>
<td>Unable to read 'method' HTTP variable.</td>
<td></td>
</tr>
<tr>
<td>10-0004</td>
<td>Unable to read 'host' HTTP variable.</td>
<td></td>
</tr>
<tr>
<td>10-0005</td>
<td>Unable to read 'URI' HTTP variable.</td>
<td></td>
</tr>
<tr>
<td>10-0007</td>
<td>URL too long.</td>
<td>Increase the MaxUriSize parameter; the default setting is 4096 bytes.</td>
</tr>
<tr>
<td>10-0001</td>
<td>Unable to read 'SERVER_NAME' HTTP variable.</td>
<td>Check that the Web browser and Web server are HTTP 1.0-compliant.</td>
</tr>
<tr>
<td>10-0002</td>
<td>Unable to read 'URL' HTTP variable.</td>
<td></td>
</tr>
</tbody>
</table>

**Table 3. SiteMinder password services error code**

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Meaning</th>
<th>Action to Take</th>
</tr>
</thead>
<tbody>
<tr>
<td>30-0026</td>
<td>Password Services redirect URL is not available.</td>
<td>Check that you have configured the redirect URL for password services.</td>
</tr>
</tbody>
</table>
### Table 4. SiteMinder communication error codes

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Meaning</th>
<th>Action to Take</th>
</tr>
</thead>
</table>
| 20-0001    | Unable to reach SiteMinder accounting server or an unexpected Policy Server error occurred. | • Check Policy Server logs for more information on the error.  
• Check connectivity between the Web Agent and the Policy Server by pinging the Policy Server.  
If a firewall is configured between the Agent and the Policy Server, check that it is not blocking the appropriate service port: **accounting**: 44441, **authentication**: 44442, **authorization**: 44443. |
| 20-0002    | Unable to reach SiteMinder authentication server or an unexpected Policy Server error occurred. |                                                                                                                                         |
| 20-0003    | Unable to reach SiteMinder authorization server or an unexpected Policy Server error occurred. |                                                                                                                                         |

### Table 5. HTTP miscellaneous 500 server error codes

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Meaning</th>
<th>Action to Take</th>
</tr>
</thead>
<tbody>
<tr>
<td>00-0001</td>
<td>Unable to resolve agent name from IP address.</td>
<td>Check the agent configuration to be sure that each HOST address served by the web server has a corresponding AgentName mapped to it or that DefaultAgentName is set properly.</td>
</tr>
</tbody>
</table>
| 00-0002    | Illegal Characters in URL. | Characters defined in BadUrlChars have been detected in a URL.  
Either remove the offending characters from the URL or remove the characters from the BadUrlChars setting so they will no longer be blocked. |
<table>
<thead>
<tr>
<th>Error Code</th>
<th>Meaning</th>
<th>Action to Take</th>
</tr>
</thead>
<tbody>
<tr>
<td>00-0003</td>
<td>Cookie received from an IP Address other than the one to which it was issued.</td>
<td>Set <code>PersistentIPCheck</code> or <code>TransientIPCheck</code> to <code>no</code> as appropriate if the failed functionality is desired. Otherwise, no action is necessary because the agent blocked an access attempt using a stolen cookie.</td>
</tr>
<tr>
<td>00-0004</td>
<td>SSLCRED cookie contains a status of error.</td>
<td>Investigate the Web Agent acting as the secure credential collector (SCC) and verify its configuration. Typically, this error only occurs when the SCC agent cannot acquire credentials from its environment, indicating a possible configuration error.</td>
</tr>
<tr>
<td>00-0005</td>
<td>FORMCRED cookie contains a status of error.</td>
<td>Investigate the Web Agent acting as the forms credential collector (FCC) and verify its configuration. Typically, this error only occurs when the FCC agent cannot acquire credentials from its environment, indicating a possible configuration error.</td>
</tr>
<tr>
<td>00-0006</td>
<td>NTLM Protected Resource not found in resource cache as expected.</td>
<td>Investigate the NTLM authentication scheme setup to verify the configuration.</td>
</tr>
<tr>
<td>00-0007</td>
<td>ASCII encoding error.</td>
<td>This is an internal Web Agent error. Investigate the Web Server and Web Agent to diagnose possible service instability. Contact Netegrity Customer Support with the web agent log and configuration files available for review.</td>
</tr>
<tr>
<td>Error Code</td>
<td>Meaning</td>
<td>Action to Take</td>
</tr>
<tr>
<td>------------</td>
<td>------------------------------</td>
<td>--------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>00-0008</td>
<td>SSL Authentication failed.</td>
<td>This error indicates a bad certificate or that the user is not authenticated.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Try a different certificate or investigate the SSL authentication scheme configuration for possible issues.</td>
</tr>
<tr>
<td>00-0009</td>
<td>Bad or Missing SSL credentials.</td>
<td>Try a different certificate or username/password pair. Investigate the SSL authentication scheme configuration for possible issues.</td>
</tr>
<tr>
<td>00-0010</td>
<td>Access Denied.</td>
<td>This error indicates a general failure that resulted in blocked access.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Investigate the Web Agent and Policy Server logs to determine the root cause of the failure.</td>
</tr>
<tr>
<td>00-0011</td>
<td>Credential Collector Error</td>
<td>This indicates a general failure in Forms or SSL based advanced authentication resulted in blocked access.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Investigate Web Agent and Policy server logs to determine the root cause of the failure.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Also, investigate the advanced authentication scheme setup for issues.</td>
</tr>
<tr>
<td>00-0012</td>
<td>Encryption Error.</td>
<td>This indicates an Internal Web Agent error.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Investigate the Web Server and Web Agent to diagnose a possible service instability.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Also, review Key Store setup to verify that proper Agent Keys are in use.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Contact Netegrity Customer Support with the web agent log and configuration files available for review.</td>
</tr>
</tbody>
</table>
### Troubleshooting Policy Server settings

1. **Configuring full logoff**

   If Webtop does not invalidate the session after users log out of Webtop while using SSO authentication, the user can configure full logoff in the Policy Server.

   You must use the `LogOffURI` parameter and create a logoff page for configuring full logoff.

   The logoff page is required if the user's browser is configured to cache pages. In this case, an HTML logoff page may be loaded from the cache and not from the Web server. As a result, the Web Agent fails to log off the user. Set up the logoff page so it cannot be cached in the user's browser.

   **To implement full logoff and use a logoff page you create:**
   
   a. Create a custom HTTP application to log off the user. For example, add an **Exit** or **Sign Off** button that redirects the user to a specific URL.
   
   b. Configure the `LogOffURI` parameter as follows:

<table>
<thead>
<tr>
<th>Error Code</th>
<th>Meaning</th>
<th>Action to Take</th>
</tr>
</thead>
</table>
| 00-0013    | Agent Configuration Error        | One or more errors occurred during startup preventing valid configuration of the agent.  
  - On Windows, check the Application Event Log for more information.  
  - For apache agents, check the apache error log for more information.  
  - For iPlanet Unix agents, start iPlanet from a shell prompt and look for possible errors displayed there through STDERR.  
  
  Check that `SmHost.conf` file exists (host is registered properly) and contains valid entries.  
  Check that `WebAgent.conf` file contains a valid `HostConfigFile` entry that points to a valid `SmHost.conf` file.  
  Check that `AgentConfigObject` contains a valid value. |
i. Add the **LogOffURI** parameter in the SiteMinder Web Agent Configuration dialog box.

ii. Enter the URI of the custom logoff HTTP page or OOTB Webtop logoff page that will log off the user. Do not enter a fully qualified URL. **For example:**

Parameter Name: LogoffUri

Parameter Value: /<context-root-name>/wdk/dologout.jsp (Ex: /webtop/wdk/dologout.jsp)

The LogOffURI parameter directs the Web Agent to the log off page and deletes the session and the authentication cookies from the browser.

c. After clearing the cookies, the Web Agent calls the Policy Server and instructs the Policy Server to remove any session information. As a result, the user is completely logged off.

The following figure illustrates how full logoff is configured in the Policy Server.

**Figure 3. Configuring full logoff in the Policy Server**
2. **Configuring session timeouts**

The user can configure session timeouts in the SiteMinder Realm dialog box. The following timeout settings are available:

**Maximum Timeout Enabled**

If this setting is defined, the values specified in the associated Hours and Minutes fields determine the maximum amount of time a user session can be active before the agent challenges the user to re-authenticate. This setting is enabled by default. The default maximum session length is two hours.

**Idle Timeout Enabled**

This setting is enabled by default. To specify no session idle timeout, clear the checkbox. The default session idle timeout is one hour. The session expires after the specified idle timeout value has lapsed.

The best practice recommended is to ensure that the values of the session expiry time of the application server and the session expiry time of the SSO cookie are identical. If business policy does not allow such a setting, we suggest that you set the application server timeout to be less than or equal to the SSO cookie timeout value.

**Example Settings:**

- Maximum Timeout Enabled = 3 hours
- Idle Timeout Enabled = 1 hour
- App Server Session Timeout = 40 minutes

The following figure illustrates the configuration of session timeouts in the Policy Server.
Known issues

- **Webtop does not invalidate the session after users log out of Webtop while using SSO authentication (WDK-2351)**

The Netegrity Policy Server failed to remove the SiteMinder session information when the user logged out of the Webtop application. You must configure full logoff in SiteMinder, which directs the Web Agent to delete the session and authentication cookies from the browser when the user logs out and instructs the Policy Server to remove any SiteMinder session information and ensure that the user is completely logged off. Full logoff is configured using the parameter "LogoffUri". The value of the "LogoffUri" parameter could be the request URI of Webtop logout page.
Add the following parameter in the SiteMinder Webtop Agent Configuration dialog box:

Parameter Name: LogoffUri
Parameter Value: /<context-root-name>/wdk/dologout.jsp (Ex: /webtop/wdk/dologout.jsp)

- **Configuring Webtop with SSO and UserPrincipalAuthentication (WEBTOP-20696)**
  The SSO and UserPrincipalAuthentication do not work and the following error occurs:


  Change the Password property name format from "password" to "new-pw" in the TrustedAuthenticatorCredentials.properties file.

  **Example format:**

  ```
  docbasename.user=cs65sp2
  docbasename.new-pw=qclvhJvdglI=
  docbasename.domain=
  ```

  The configuration options do not work as per the documentation. The .password fields are not compatible with Documentum Content Server 6.5 SP2, 6.5 SP3, and 6.6. Therefore, you must use the “docbasename.new-pw=" format for these releases.

- **UCF Client is unable to connect to the server with SSO when using DRL (WEBTOP-21132)**
  A user can encounter sporadic errors as follows, while accessing UCF content transfer using the Webtop DRL. If they have configured Tomcat with SiteMinder:

  Failed to connect to server - refer ucf client logs for details
  com.documentum.ucf.client.install.launcher.InvocationException: An error has occurred when invoking the UCF runtime. Details:
  Failed to connect to server - refer ucf client logs for details at com.documentum.ucf.client.install.launcher.impl.RuntimeInvoker.retrieveContent(RuntimeInvoker.java:603)
When the error occurs, the SMSESSION parameter is empty in the UCF Client log. The error occurs because the cookies are enclosed within double quotes introduced by the Tomcat application server. Perform the following change in the Tomcat configuration to remove the double quotes:

File: catalina.properties

Add the following property:
org.apache.catalina.STRICT_SERVLET_COMPLIANCE=true

If you set the value to “true”, cookies are parsed strictly; by default v0 cookies will not work with any invalid characters.

If the value is set to “false”, any v0 cookie with an invalid character will be switched to a v1 cookie and the value will be enclosed within double quotes. The default value is “false”.

**Conclusion**

This white paper captures information used to troubleshoot SiteMinder SSO issues with the Webtop application. We have explained how Netegrity SiteMinder authentication works with the Webtop application; settings to configure on Webtop, Web Agent, Content Server, and Policy Server; how to troubleshoot Web Agent issues and interpret Web Agent error codes; and the log files required to capture across all layers of the deployment topology.

This paper has also briefly explained troubleshooting steps to resolve SiteMinder SSO issues with Webtop.