CA GREEN PAPERS

CA Clarity™ PPM

Integrating with CA SiteMinder®

Transforming IT Management

LEGAL NOTICE

This publication is based on current information and resource allocations as of its date of publication and is subject to change or withdrawal by CA at any time without notice. The information in this publication could include typographical errors or technical inaccuracies. CA may make modifications to any CA product, software program, method or procedure described in this publication at any time without notice.

Any reference in this publication to non-CA products and non-CA websites are provided for convenience only and shall not serve as CA's endorsement of such products or websites. Your use of such products, websites, and any information regarding such products or any materials provided with such products or at such websites shall be at your own risk.

Notwithstanding anything in this publication to the contrary, this publication shall not (i) constitute product documentation or specifications under any existing or future written license agreement or services agreement relating to any CA software product, or be subject to any warranty set forth in any such written agreement; (ii) serve to affect the rights and/or obligations of CA or its licensees under any existing or future written license agreement or services agreement relating to any CA software product; or (iii) serve to amend any product documentation or specifications for any CA software product. The development, release and timing of any features or functionality described in this publication remain at CA's sole discretion.

The information in this publication is based upon CA's experiences with the referenced software products in a variety of development and customer environments. Past performance of the software products in such development and customer environments is not indicative of the future performance of such software products in identical, similar or different environments. CA does not warrant that the software products will operate as specifically set forth in this publication. CA will support only the referenced products in accordance with (i) the documentation and specifications provided with the referenced product, and (ii) CA's then-current maintenance and support policy for the referenced product.

Certain information in this publication may outline CA's general product direction. All information in this publication is for your informational purposes only and may not be incorporated into any contract. CA assumes no responsibility for the accuracy or completeness of the information. To the extent permitted by applicable law, CA provides this document "AS IS" without warranty of any kind, including, without limitation, any implied warranties of merchantability, fitness for a particular purpose, or non-infringement. In no event will CA be liable for any loss or damage, direct or indirect, from the use of this document, including, without limitation, lost profits, lost investment, business interruption, goodwill or lost data, even if CA is expressly advised of the possibility of such damages.

COPYRIGHT LICENSE AND NOTICE:

This publication may contain sample application programming code and/or language which illustrate programming techniques on various operating systems. Notwithstanding anything to the contrary contained in this publication, such sample code does not constitute licensed products or software under any CA license or services agreement. You may copy, modify and use this sample code for the purposes of performing the installation methods and routines described in this document. These samples have not been tested. CA does not make, and you may not rely on, any promise, express or implied, of reliability, serviceability or function of the sample code.

TITLE AND PUBLICATION DATE:

CA Clarity ™ PPM Integrating with CA SiteMinder®

Publication Date: February 26, 2010

ACKNOWLEDGEMENTS

Principal Authors and Technical Editors

Sean Harp
Josh Leone
Jack Saunders

Editor
Pamela Molennor

The principal authors, editors and CA would like to thank the following teams:

CA Services
Development
Marketing
QA
Support
SWAT
Technical Sales
Technical Information

CA PRODUCT REFERENCES

This document references the following CA products:

- CA Clarity™ PPM r8.1.2 or higher
- CA SiteMinder® r6 SP5 or higher

Note: Always refer to the latest CA Clarity PPM Product Architecture Stack (PAS) for product details. It is available on CA Support Online (http://support.ca.com) with the CA Clarity PPM product documentation.

THIRD-PARTY ACKNOWLEDGEMENTS

- IBM® WebSphere®

  IBM and WebSphere are trademarks of International Business Machines Corporation in the United States, other countries, or both.

- Oracle® WebLogic

  Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

- Apache Tomcat™

  Apache and Apache Tomcat are trademarks of The Apache Software Foundation.

FEEDBACK

Please email us at greenbooks@ca.com to share your feedback on this publication. Please include the title of this publication in the subject of your email response. For technical assistance with a CA product, please contact CA Technical Support at http://ca.com/support. For assistance with support specific to Japanese operating systems, please contact CA at http://www.casupport.jp.

Contents

Chapter 1: Introduction

Who Should Read This Paper?

This paper provides the software architect, software developer, software engineer, system administrator, and support technician with the information necessary to integrate CA Clarity PPM with CA SiteMinder. It is intended for technically-oriented people who require an advanced level of understanding of the CA Clarity PPM integration capabilities to successfully configure and maintain their CA Clarity PPM environment.

What is CA SiteMinder?

CA SiteMinder provides a centralized security management foundation that enables user authentication and controlled access to web applications and portals. CA SiteMinder delivers advanced security management capabilities and enterprise-class site administration, enabling greater IT control and security.

CA SiteMinder provides the following features:

- Single Sign-On (SSO)
- Strong Authentication Management
- Centralized Policy-Based Authorization and Audit
- Identity Federation
- Enterprise Manageability

CA SiteMinder integrates with industry-leading directory services and user stores, eliminating redundant administration of user information. This integration simplifies administration and provides unique and comprehensive security capabilities. CA SiteMinder fully leverages existing user directories, from leading LDAP directories and relational databases to mainframe security directories.

Integration Points and Functionality

The integration protects the CA Clarity PPM application URIs with CA SiteMinder by creating several realms, rules, and a policy. A rule identifies and controls access to specific resources that are included in the policy. A CA SiteMinder policy binds rules and responses to users, groups and roles. The responses in a policy enable the solution to customize the delivery of content for each user.

Policies reside in the policy store, which is the data source that contains all of the CA SiteMinder entitlement information. When a CA Clarity PPM user tries to access the protected URI, that is, /niku/* URI and is not already authenticated, CA SiteMinder displays a login window and challenges the user for credentials. The user enters credentials and submits them, and is then authenticated against the CA SiteMinder Policy server. CA SiteMinder sets the SM_USER HTTP Header and redirects the request to the CA Clarity

PPM Overview URL, so the CA Clarity PPM application is available to users to start their work.

The following steps provide an overview of how the CA Clarity PPM and CA SiteMinder integration works:

1. The user attempts to access a protected resource, which is the CA Clarity PPM web application.
2. The user is challenged for credentials and presents them to the Web Agent.
3. The user's credentials are passed to the policy server.
4. The user is authenticated against the appropriate user store (such as LDAP or Active Directory).
5. The user receives access to the secured CA Clarity PPM web application.

Introduction to the CA SiteMinder Integration to CA Clarity PPM

Prior to the release of CA Clarity PPM 8.1.2, integration with CA SiteMinder for single sign-on (SSO) required that at least one additional non-SSO instance of the CA Clarity PPM application be configured for use with solutions that integrate with CA Clarity PPM or its non-SSO components.

An enhancement included in the release of CA Clarity PPM 8.1.2 now permits XOG and the schedulers to successfully authenticate against an SSO-enabled CA Clarity PPM application instance without including a valid SSO token with the request. This configuration requires some modifications be made to the CA SiteMinder Policy server so that XOG and the schedulers can bypass the CA SiteMinder Web Agent protection.

This document serves as a reference for the CA SiteMinder Policy Server changes necessary to properly integrate with CA Clarity PPM 8.1.2 or higher.

Chapter 2: Configure for SSO

Several changes must be made to CA Clarity PPM via the CSA (Clarity System Administrator) in order to enable single sign-on (SSO). The configuration process and considerations are described in the following section.

Configure the SSO Token in the Security Section

In the CSA, navigate to your server Properties and click on the *Security* link to configure the security properties (the tab highlighted in white below). Configure the Single Sign-on options and click Save. The CA Clarity PPM services will need to be restarted before these changes will take effect.

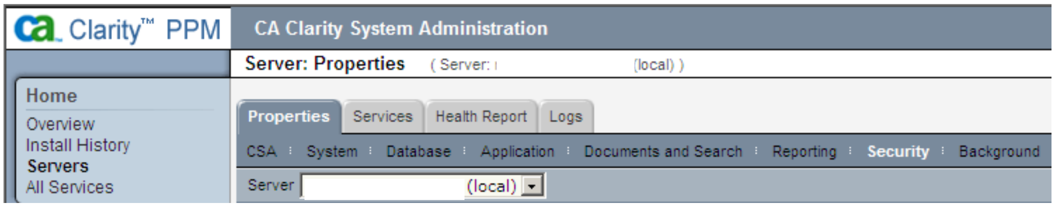
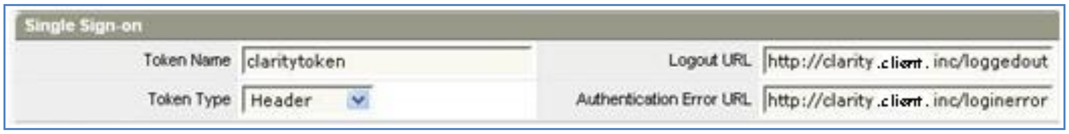


Figure 1: CSA, Security Section

The last section on the *Security* page is labeled *Single Sign-on*.



Single Sign-on

Token Name: claritytoken

Token Type: Header

Logout URL: http://clarity.client.inc/loggedout

Authentication Error URL: http://clarity.client.inc/loginerror

Figure 2: CSA, Single Sign-on

Token Name

When SiteMinder is configured for CA Clarity PPM, a custom response will be added that will set an HTTP header (named **claritytoken** in the examples presented in this document) with the user name from the LDAP Directory used for authentication. This token can be named anything, but the name must match the token name that is configured in SiteMinder.

Token Type

The *Token Type* should be set to **Header** so that CA Clarity PPM checks the HTTP headers for the SiteMinder token.

Logout URL

Configure the logout URL with the location you would like the end user redirected to when they click the Logout link within CA Clarity PPM or when either their CA Clarity PPM or SiteMinder sessions expire.Authentication Error URL

Configure this URL with the location to redirect the user if any authentication error is encountered.

Configure Application Bind Address and Port

In the CSA, navigate to your server Properties and select the *Application* link (the tab highlighted in yellow in the screenshot below) to configure the Apache Tomcat application server. The bind addresses and port configurations in the CSA do not pertain to IBM WebSphere and Oracle WebLogic configurations, which are addressed in later paragraphs.

Within the **app** instance settings, check the box next to *Use Single Sign-on*. An example of this is shown in Figure 4.

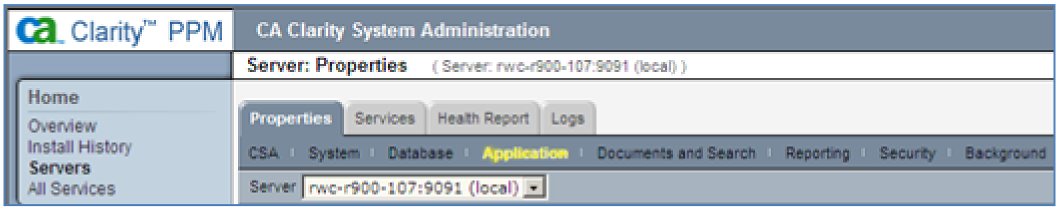


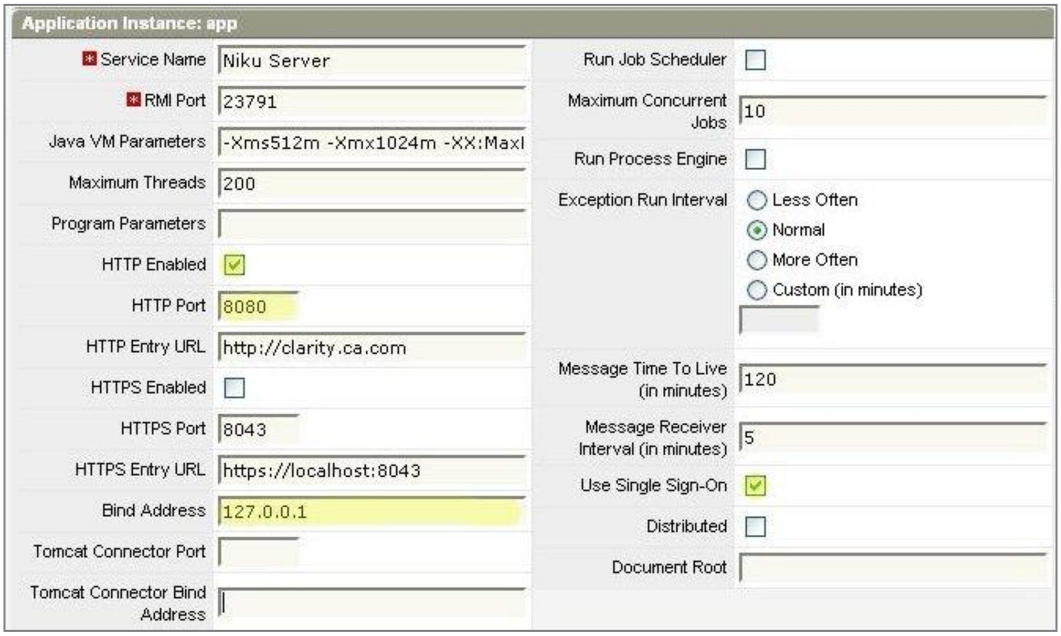
Figure 3: CSA, Application Section

Proxy from Apache HTTP Server with mod_proxy

If using mod_proxy (simple HTTP reverse proxy), configure the *HTTP Port* for the **app** instance to a non-privileged port (8080 will be used as an example in Chapter 4: Proxy from Apache HTTP Server).

Also configure the *Bind Address* field for the **app** instance (not the *Tomcat Connector Bind Address)* to the loopback address (127.0.0.1) if the Apache HTTP server is on the same host.

The following screen shot illustrates this configuration:



Application Instance: app

Service Name	Niku Server	Run Job Scheduler	□
RMI Port	23791	Maximum Concurrent Jobs	10
Java VM Parameters	-Xms512m -Xmx1024m -XX:MaxI	Run Process Engine	□
Maximum Threads	200	Exception Run Interval	○ Less Often ● Normal ○ More Often ○ Custom (in minutes)
Program Parameters			
HTTP Enabled	☑		
HTTP Port	8080		
HTTP Entry URL	http://clarity.ca.com	Message Time To Live (in minutes)	120
HTTPS Enabled	□		
HTTPS Port	8043	Message Receiver Interval (in minutes)	5
HTTPS Entry URL	https://localhost:8043	Use Single Sign-On	☑
Bind Address	127.0.0.1	Distributed	□
Tomcat Connector Port		Document Root	
Tomcat Connector Bind Address			

Figure 4: CSA, Application Properties - Reverse Proxy

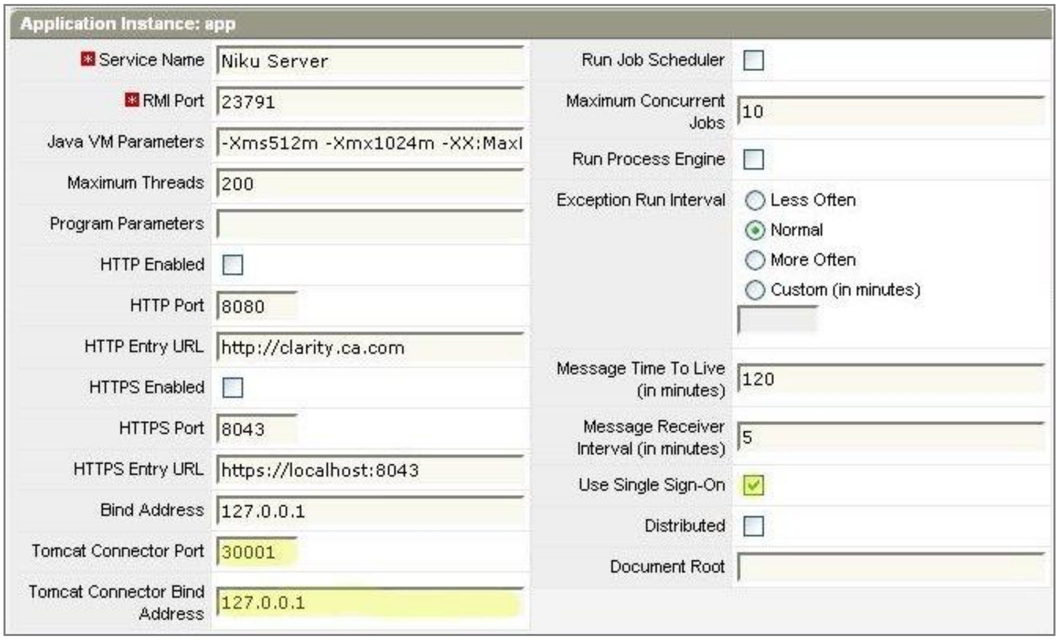
Proxy from Microsoft IIS or Apache HTTP Server with the Apache Tomcat Connector

The Apache Tomcat Connector requires that the *Tomcat Connector Port* and *Tomcat Connector Bind Address* for the CA Clarity PPM **app** instance be configured. The Apache Tomcat Connector utilizes the **ajp13** protocol and must be configured to connect to the *Tomcat Connector Port* and not the *HTTP Port*.

Set the *Tomcat Connector Port* to a non-privileged port (30001 is used in the examples).

Set the *Tomcat Connector Bind Address* to the loopback address (127.0.0.1) if the Microsoft IIS or Apache HTTP server is on the same host.

The following screen shot illustrates this configuration:



Application Instance: app

Service Name	Niku Server	Run Job Scheduler	□
RMI Port	23791	Maximum Concurrent Jobs	10
Java VM Parameters	-Xms512m -Xmx1024m -XX:MaxI	Run Process Engine	□
Maximum Threads	200	Exception Run Interval	○ Less Often ● Normal ○ More Often ○ Custom (in minutes)
Program Parameters			
HTTP Enabled	□		
HTTP Port	8080		
HTTP Entry URL	http://clarity.ca.com	Message Time To Live (in minutes)	120
HTTPS Enabled	□		
HTTPS Port	8043	Message Receiver Interval (in minutes)	5
HTTPS Entry URL	https://localhost:8043	Use Single Sign-On	☑
Bind Address	127.0.0.1	Distributed	□
Tomcat Connector Port	30001	Document Root	
Tomcat Connector Bind Address	127.0.0.1		

Figure 5: CSA, Application Properties - Tomcat Connector

Firewall Considerations for Multiple Host Configuration

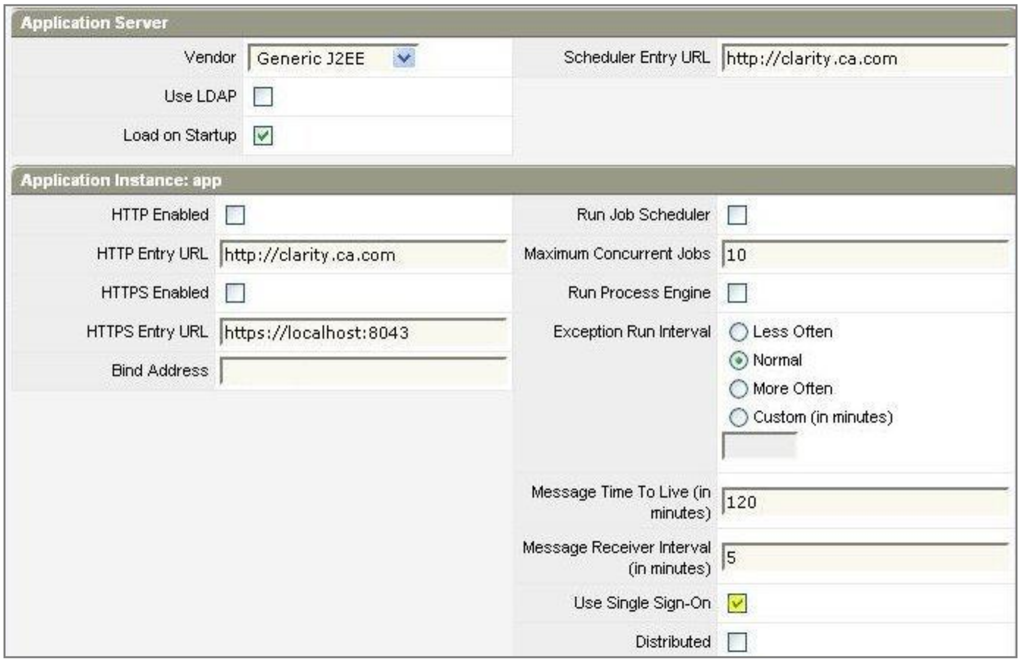
If the Microsoft IIS or Apache HTTP server proxy into CA Clarity PPM resides on a separate physical host other than the CA Clarity PPM application instance, configure firewall rules (IPtables for Linux, Windows® Firewall, or hardware firewall) so that only the HTTP proxy server IP address is permitted to reach the bind address and port chosen in the steps above. Refer to your operating system documentation for information on firewall rule settings.

Failure to secure Tomcat Application bind address leaves the CA Clarity PPM application open to attacks that bypass the SiteMinder Web Agent protection.

IBM WebSphere and Oracle WebLogic Considerations

Enable SSO in the Application Properties

In a CA Clarity PPM J2EE configuration utilizing IBM WebSphere or Oracle WebLogic, the bind address and port settings are not present in the CSA Application Properties. The checkbox for "Use Single Sign-on" must be checked to enable SSO.



Application Server

Vendor	Generic J2EE	Scheduler Entry URL	http://clarity.ca.com
Use LDAP	□		
Load on Startup	☑		

Application Instance: app

HTTP Enabled	□	Run Job Scheduler	□
HTTP Entry URL	http://clarity.ca.com	Maximum Concurrent Jobs	10
HTTPS Enabled	□	Run Process Engine	□
HTTPS Entry URL	https://localhost:8043	Exception Run Interval	○ Less Often ● Normal ○ More Often ○ Custom (in minutes)
Bind Address			
		Message Time To Live (in minutes)	120
		Message Receiver Interval (in minutes)	5
		Use Single Sign-On	☑
		Distributed	□

Figure 6: Enabling SSO

Stand-alone WebSphere or WebLogic Instances

Both WebSphere and WebLogic typically have dedicated web servers that utilize a proxy plug-in. These web servers are typically based on Apache HTTP Server, but instead of utilizing mod_proxy their respective custom plug-ins provide the mechanism for proxy into the application tier.

Bind addresses and ports, in this case, would be set within either the WebSphere or WebLogic application consoles to ensure that no end-user is able to bypass the web tier and the SiteMinder Web Agent. There they can be set to the loopback interface (127.0.0.1) to prevent end-users from bypassing the SiteMinder Web Agent.

Websphere or WebLogic Clusters

When utilizing a WebSphere or WebLogic cluster that spans multiple hosts, it will not be possible to bind the application instances to the loopback interface. In these configurations, the vendor-supplied proxy plug-ins must be able to reach all of the application cluster members for failover purposes.

In this type of configuration the application instances would need to be bound to public IP addresses. Use the software firewall provided, with the operating system or a hardware firewall, to prevent end-users from bypassing the SiteMinder Web Agent and hitting the application instances directly.

LDAP Configuration in CA Clarity PPM for Schedulers

If either of the scheduler applications (Open Workbench and Microsoft Office Project) will be used with the CA Clarity PPM application, it is recommended that you configure CA Clarity PPM to authenticate against the same LDAP directory that is used by SiteMinder for authentication. After a period of inactivity in these scheduler applications, the CA Clarity PPM session in use will become invalid. When this occurs, Microsoft Office Project and Open Workbench will prompt the user to authenticate before they can continue working. If LDAP is not configured as the authentication source for CA Clarity PPM, this authentication will fail.

Note: Please refer to the *CA Clarity PPM Installation Guide* for a detailed description of the steps for configuring CA Clarity PPM to authenticate with LDAP.

Chapter 3: Policy Server Configuration

Overview of URI Patterns

Typically in a CA SiteMinder/CA Clarity PPM integration, SiteMinder protects the URI /niku/* and the SiteMinder configuration is relatively simple. The configuration becomes a little more complex when unprotecting the URIs used by the CA Clarity PPM schedulers, because the URI context for the schedulers is shared with that used by the browser-based users.

Example:

/niku/app?action=npt.overview (used by browser-based users)

/niku/app?action=schedulers.login (used by the schedulers).

In order to protect all /niku/app URI patterns used by the browser-based users, but leave unprotected the URI patterns used by the schedulers, we must use a regular expression to specifically protect any /niku/app* URI that is not used by the schedulers.

In addition to handling the scheduler URIs, now that we are required to not simply protect /niku/*, we must add a few additional realms to protect other URIs used by CA Clarity PPM:

/niku/gantt*	(used by the CA Clarity PPM Gantt functionality)
/niku/proxool	(a servlet for viewing connection pool statistics)

SiteMinder Realms

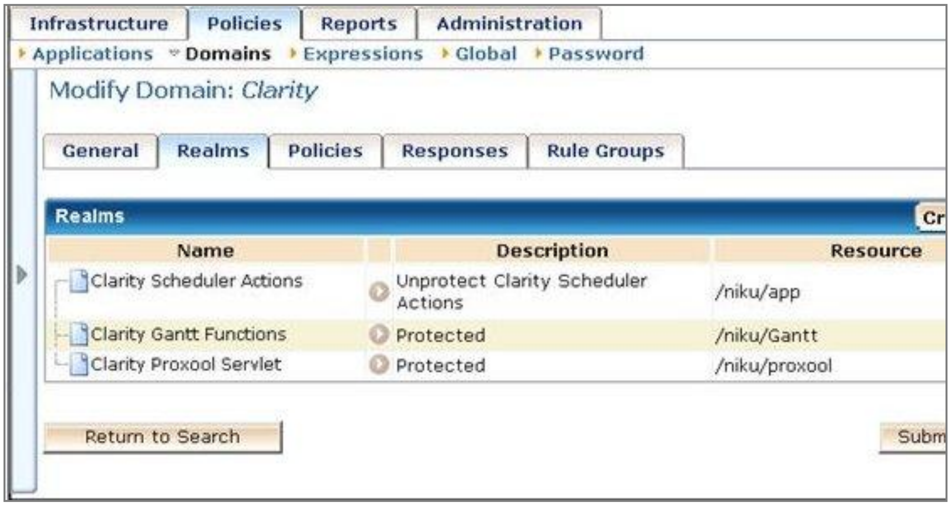
It is necessary to create three realms in SiteMinder for CA Clarity PPM:

1. CA Clarity PPM Scheduler Actions (/niku/app)

2. CA Clarity PPM Gantt Functions (/niku/Gantt)

3. CA Clarity PPM Proxool Servlet (/niku/proxool)

These realms, illustrated in the screen shot below, are discussed in the following paragraphs.



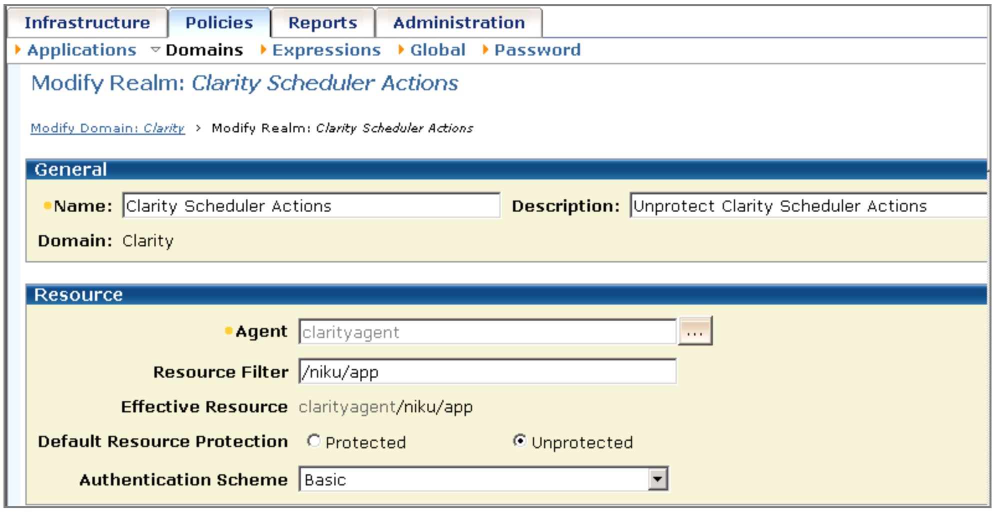
Name	Description	Resource
Clarity Scheduler Actions	Unprotect Clarity Scheduler Actions	/niku/app
Clarity Gantt Functions	Protected	/niku/Gantt
Clarity Proxool Servlet	Protected	/niku/proxool

Figure 7: SiteMinder Realms for CA Clarity PPM

Realm 1: CA Clarity PPM Scheduler Actions

This realm is the most difficult to understand. SiteMinder must be configured to unprotect every URI that does not match the pattern: */niku/app*action=sch*

All other URIs underneath /niku/app will be protected, but the URIs used by the scheduler will be allowed to pass unprotected. To accomplish this, a regular expression must be used.

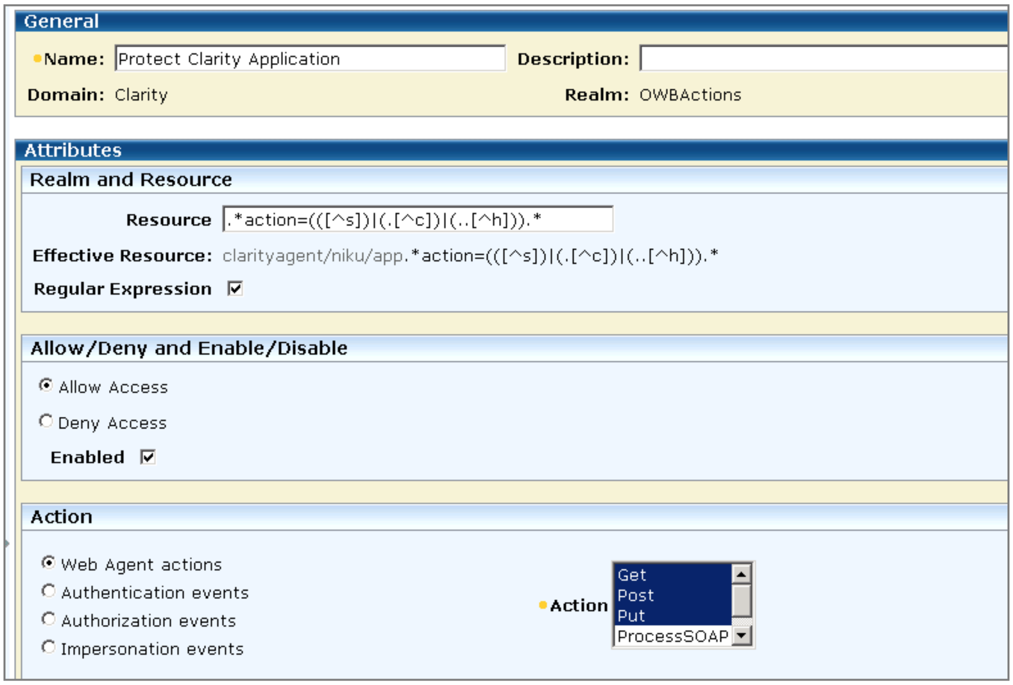

Infrastructure
Policies
Reports
Administration
Applications
Domains
Expressions
Global
Password
Modify Realm: *Clarity Scheduler Actions*
Modify Domain: *Clarity* > Modify Realm: *Clarity Scheduler Actions*
General
Resource
...

Name: Clarity Scheduler Actions
Description: Unprotect Clarity Scheduler Actions
Domain: Clarity

Agent: clarityagent
Resource Filter: /niku/app
Effective Resource: clarityagent/niku/app
Default Resource Protection: ○ Protected ● Unprotected
Authentication Scheme: Basic

Figure 8: CA Clarity PPM Scheduler Actions SiteMinder Realm

After the realm is created, create a rule under the realm that protects any URI that is not the scheduler pattern as shown here:



General

Name: Protect Clarity Application
Description:
Domain: Clarity
Realm: OWBActions

Attributes

Realm and Resource

Resource: .*action=(([^s])|(.[^c])|(..[^h])).*
Effective Resource: clarityagent/niku/app.*action=(([^s])|(.[^c])|(..[^h])).*
Regular Expression ☑

Allow/Deny and Enable/Disable

● Allow Access
○ Deny Access
Enabled ☑

Action

● Web Agent actions
○ Authentication events
○ Authorization events
○ Impersonation events

Action: Get, Post, Put (selected); ProcessSOAP

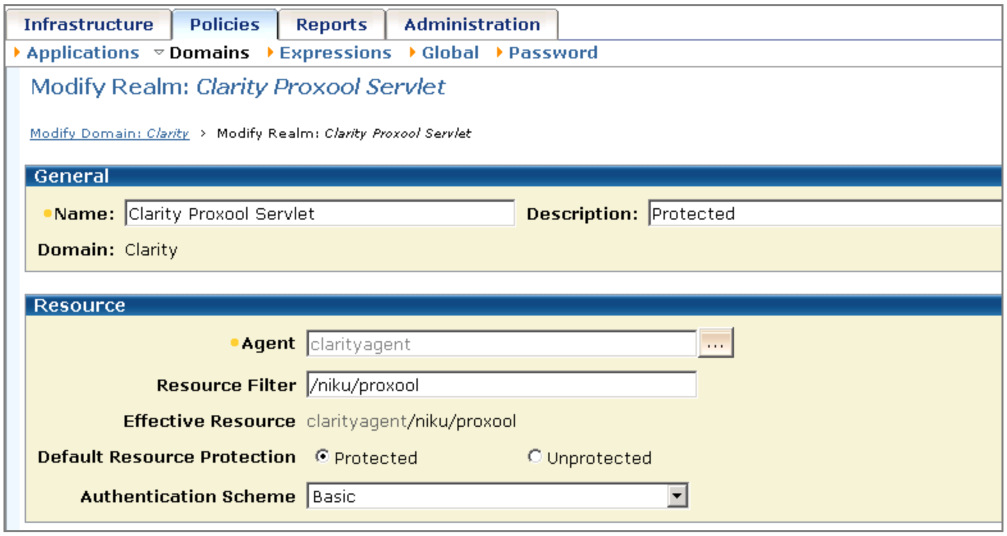
Figure 9: CA Clarity PPM Scheduler Actions SiteMinder Rule

The Resource regular expression pattern is:

```
        .*action=(([^s])|(.[^c])|(..[^h])).*
```

Realm 2: CA Clarity PPM Proxool Servlet

The Proxool servlet realm simply protects the URI /niku/proxool. This servlet provides monitoring capability in CA Clarity PPM for the JDBC database connection pool.



Modify Realm: *Clarity Proxool Servlet*

Modify Domain: *Clarity* > Modify Realm: *Clarity Proxool Servlet*

General

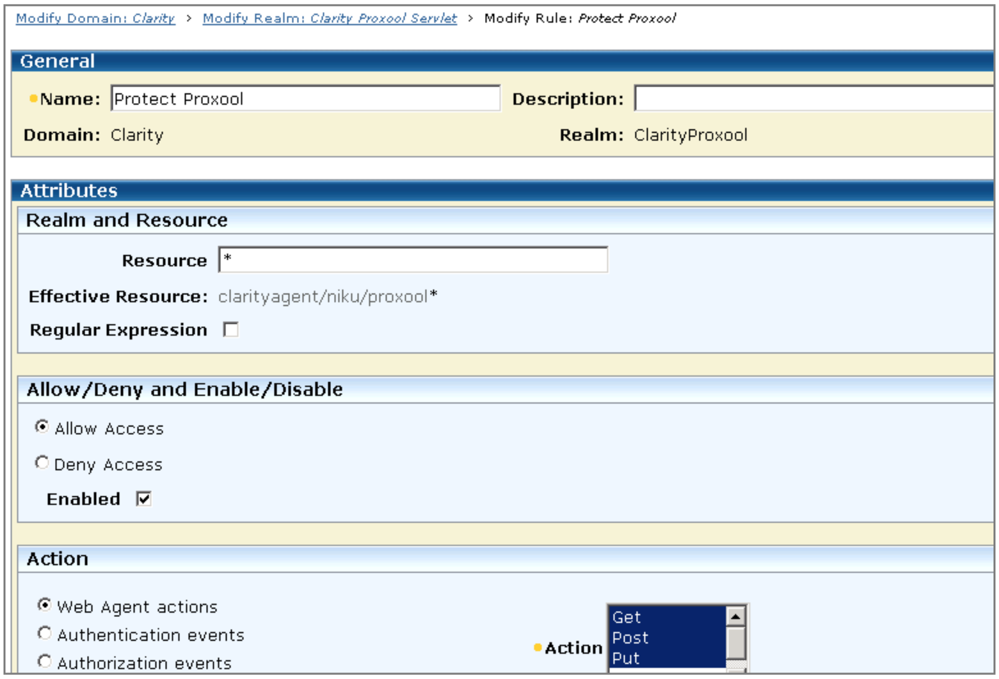
Name: Clarity Proxool Servlet
Description: Protected
Domain: Clarity

Resource

Agent: clarityagent
Resource Filter: /niku/proxool
Effective Resource: clarityagent/niku/proxool
Default Resource Protection: ● Protected ○ Unprotected
Authentication Scheme: Basic

Figure 10: CA Clarity PPM Proxool Servlet SiteMinder Realm

The rule for the Proxool servlet realm is shown here:



Modify Domain: *Clarity* > Modify Realm: *Clarity Proxool Servlet* > Modify Rule: *Protect Proxool*

General

Name: Protect Proxool **Description:**

Domain: Clarity **Realm:** ClarityProxool

Attributes

Realm and Resource

Resource *

Effective Resource: clarityagent/niku/proxool*

Regular Expression □

Allow/Deny and Enable/Disable

● Allow Access

○ Deny Access

Enabled ☑

Action

● Web Agent actions

○ Authentication events

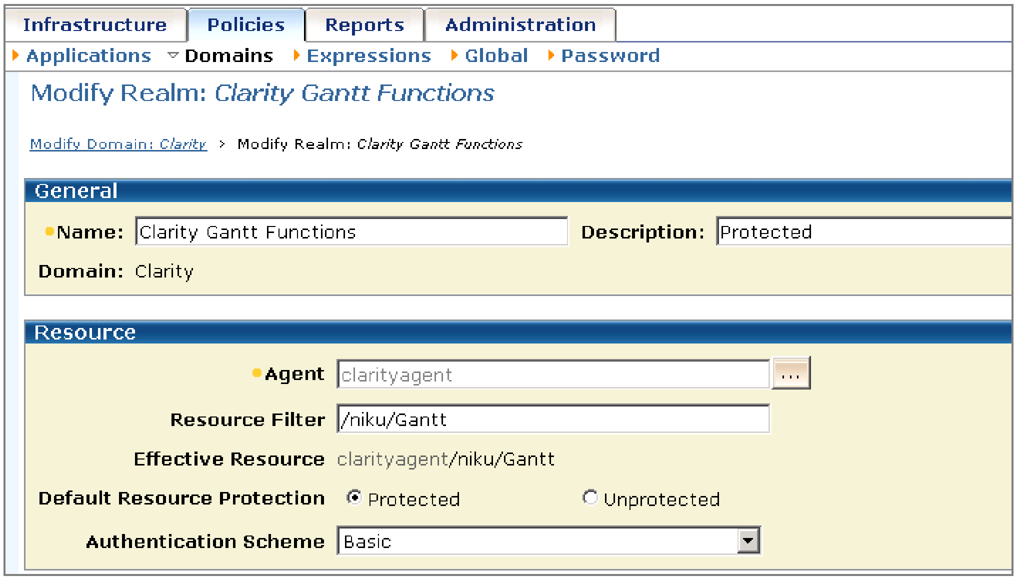
○ Authorization events

Action Get Post Put

Figure 11: CA Clarity PPM Proxool Servlet SiteMinder Rule

Realm 3: CA Clarity PPM Gantt Functions

The Gantt chart functionality in CA Clarity PPM uses the URI /niku/gantt* and must also be protected.



Modify Realm: *Clarity Gantt Functions*

Modify Domain: *Clarity* > Modify Realm: *Clarity Gantt Functions*

General

Name: Clarity Gantt Functions **Description:** Protected

Domain: Clarity

Resource

Agent clarityagent

Resource Filter /niku/Gantt

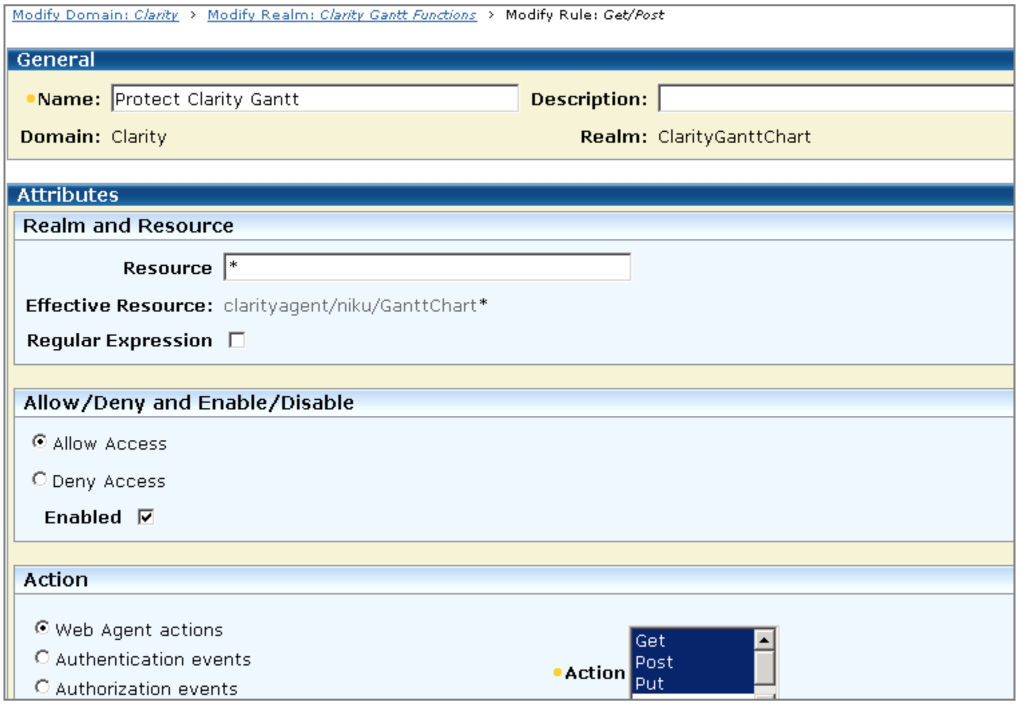
Effective Resource clarityagent/niku/Gantt

Default Resource Protection ● Protected ○ Unprotected

Authentication Scheme Basic

Figure 12: CA Clarity PPM Gantt Functions SiteMinder Realm

The rule for the Gantt functions is shown here:



Modify Domain: *Clarity* > Modify Realm: *Clarity Gantt Functions* > Modify Rule: *Get/Post*

General

Name: Protect Clarity Gantt **Description:**

Domain: Clarity **Realm:** ClarityGanttChart

Attributes

Realm and Resource

Resource *

Effective Resource: clarityagent/niku/GanttChart*

Regular Expression □

Allow/Deny and Enable/Disable

● Allow Access

○ Deny Access

Enabled ☑

Action

● Web Agent actions

○ Authentication events

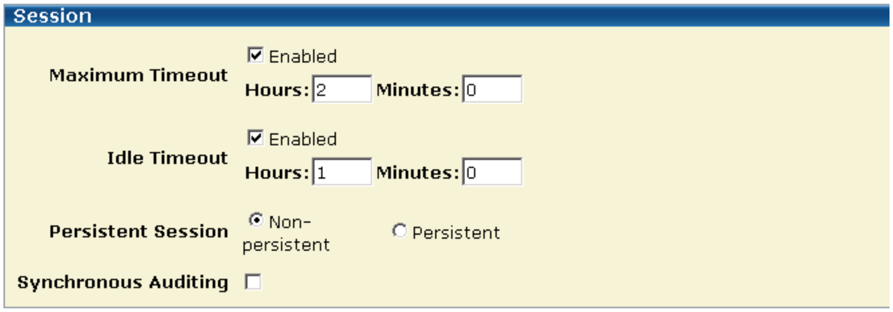
○ Authorization events

Action Get, Post, Put

Figure 13: CA Clarity PPM Gantt Functions SiteMinder Rule

Realm Idle Timeouts

Set the idle session timeout to match the CA Clarity PPM application session timeout value. Both CA Clarity PPM and CA SiteMinder default to 1 hour idle session timeout. The CA Clarity PPM session timeout is very important because it ensures that old inactive user sessions are removed from the Java heap. Set this for all of the realms that have been defined for CA Clarity PPM.



Session

Maximum Timeout ☑ Enabled **Hours:** 2 **Minutes:** 0

Idle Timeout ☑ Enabled **Hours:** 1 **Minutes:** 0

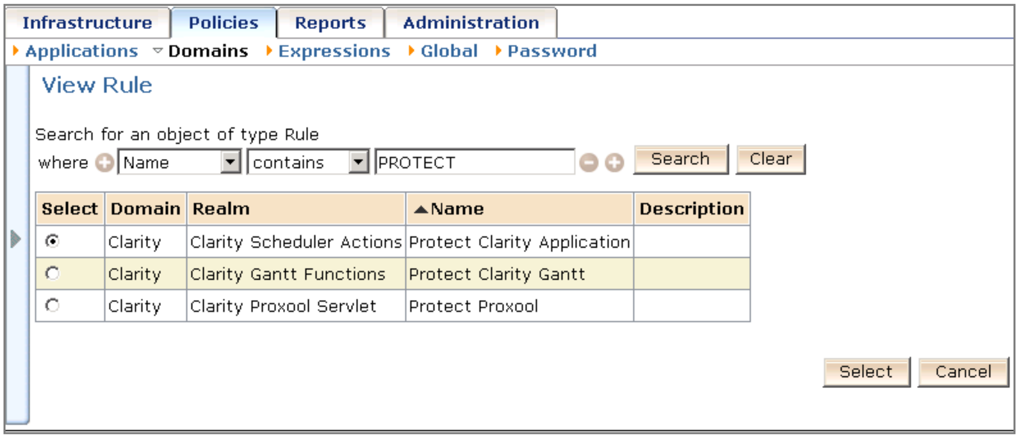
Persistent Session ● Non-persistent ○ Persistent

Synchronous Auditing □

Figure 14: SiteMinder Realm Idle Timeouts

Create SiteMinder Policy

Once the three realms and their three rules have been configured, create a policy under the Clarity SiteMinder domain and include the three rules as shown here:



Search for an object of type Rule

where Name contains PROTECT

Select	Domain	Realm	▲Name	Description
●	Clarity	Clarity Scheduler Actions	Protect Clarity Application	
○	Clarity	Clarity Gantt Functions	Protect Clarity Gantt	
○	Clarity	Clarity Proxool Servlet	Protect Proxool	

Figure 15: SiteMinder Policy - Include All Rules

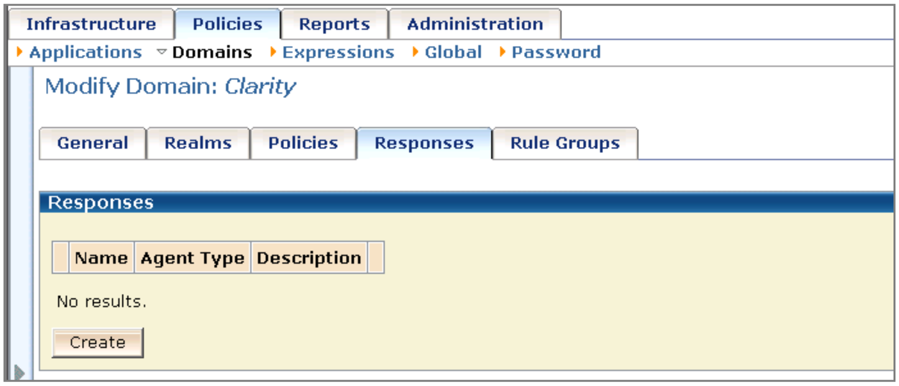
Create SiteMinder Response

To provide a consistent HTTP header token for CA Clarity PPM to use for SSO, create a response that maps the user name LDAP Directory attribute to the new HTTP header **claritytoken**. The SiteMinder built-in HTTP header for the user name is **SM_USER**, but when Microsoft IIS is configured with the Apache Tomcat Connector, all of the HTTP header variables have the underscore characters turned into dashes, making it **SM-USER**. It is easier and more consistent to simply make a token that does not contain the underscore character.

Note: The token name can be anything, but it must match the value that has been configured for SSO within CA Clarity PPM.

Follow these steps:

1. Create a new response named **Clarity Token**.



Name	Agent Type	Description

No results.

Figure 16: SiteMinder Response - Create New Response

2. Create a new attribute within the response, with the following characteristics:

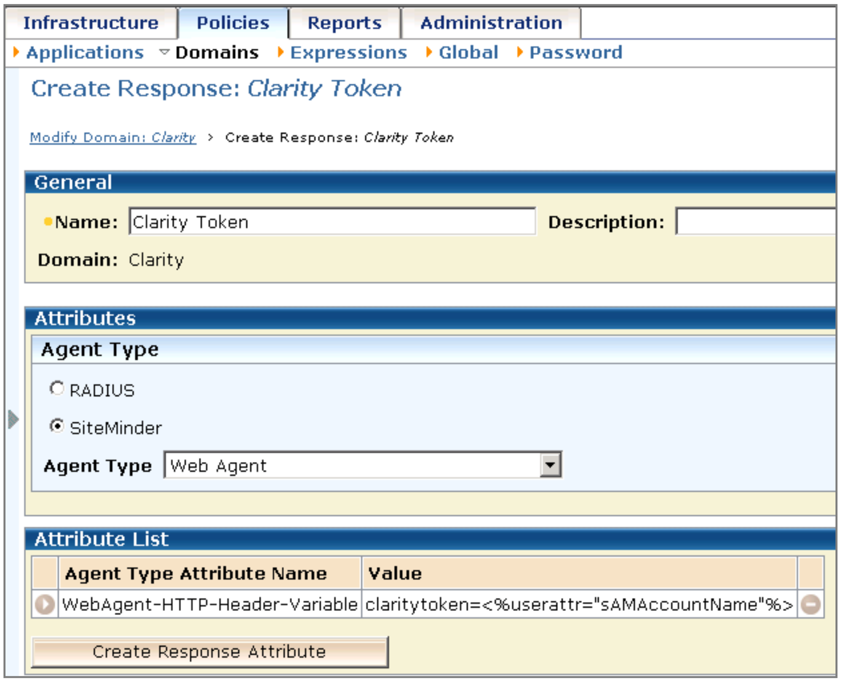
   Attribute Kind: User Attribute

   Variable Name: claritytoken

   Attribute Name: name of the LDAP user_name attribute

   The attribute name for the user name field differs between LDAP implementations and configurations. For CA Directory or other LDAP RFC compliant directories, use **uid**. For Microsoft Active Directory, use **sAMAccountName**. If CA Clarity PPM is being configured to use the email address as the login ID, use **mail** in LDAP.

   Illustrations of the response properties and the response attribute details are shown below.



Create Response: *Clarity Token*

Modify Domain: *Clarity* > Create Response: *Clarity Token*

General

Name: Clarity Token
Description:
Domain: Clarity

Attributes

Agent Type

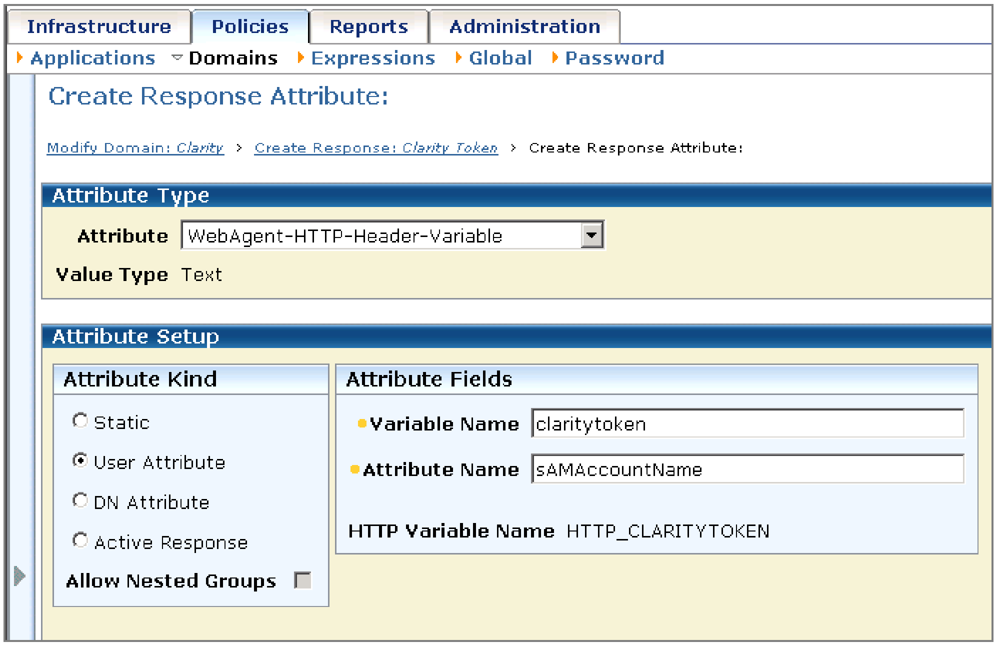
○ RADIUS
● SiteMinder

Agent Type: Web Agent

Attribute List

Agent Type Attribute Name	Value
WebAgent-HTTP-Header-Variable	claritytoken=<%userattr="sAMAccountName"%>

Figure 17: SiteMinder Response Properties



Create Response Attribute:

Modify Domain: *Clarity* > Create Response: *Clarity Token* > Create Response Attribute:

Attribute Type

Attribute WebAgent-HTTP-Header-Variable

Value Type Text

Attribute Setup

Attribute Kind

○ Static
● User Attribute
○ DN Attribute
○ Active Response

Allow Nested Groups □

Attribute Fields

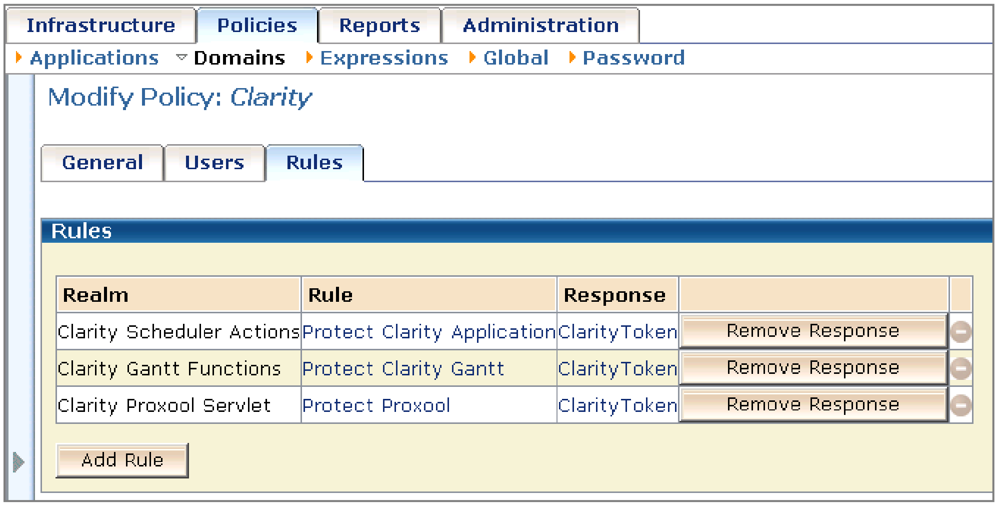
Variable Name claritytoken

Attribute Name sAMAccountName

HTTP Variable Name HTTP_CLARITYTOKEN

Figure 18: SiteMinder Response Attribute Details

3. Within the CA Clarity PPM Policy, map the Clarity Token response, created in step 2, to all of the SiteMinder rules that you created previously as shown here:



Modify Policy: *Clarity*

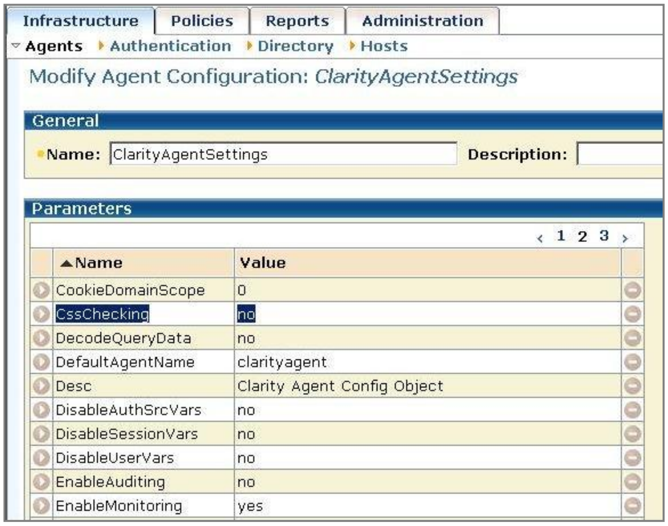
Rules

Realm	Rule	Response	
Clarity Scheduler Actions	Protect Clarity Application	ClarityToken	Remove Response
Clarity Gantt Functions	Protect Clarity Gantt	ClarityToken	Remove Response
Clarity Proxool Servlet	Protect Proxool	ClarityToken	Remove Response

Figure 19: SiteMinder Policy - Associating Response to Rules

CSS Checking and CA Clarity PPM Gantt Charts

The CA Clarity PPM Gantt functionality will trigger a BadCSS character check in the SiteMinder agent. In order for this functionality to work properly, the SiteMinder agent configuration object must have CssChecking set to "no" as shown here:



Modify Agent Configuration: *ClarityAgentSettings*

General

Name: ClarityAgentSettings Description:

Parameters

▲Name	Value
CookieDomainScope	0
CssChecking	no
DecodeQueryData	no
DefaultAgentName	clarityagent
Desc	Clarity Agent Config Object
DisableAuthSrcVars	no
DisableSessionVars	no
DisableUserVars	no
EnableAuditing	no
EnableMonitoring	yes

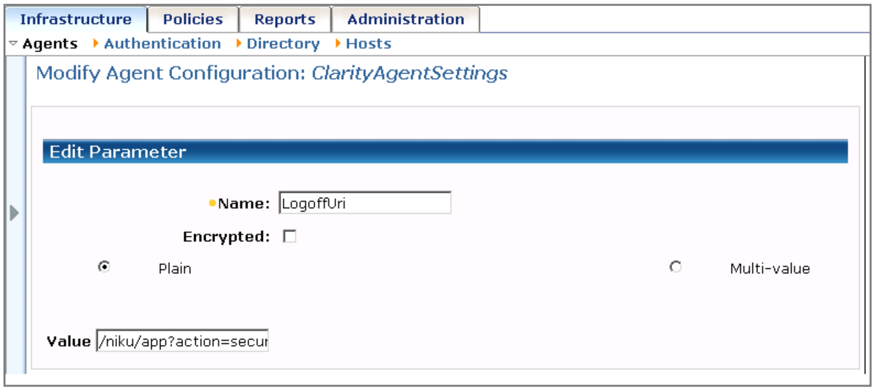
Figure 20: SiteMinder Agent Configuration for CSSchecking

Note: The “no” CssChecking setting disables cross-site script protection. Incoming URLs will not be checked for cross-site script attacks. The CA Clarity PPM application will be left to protect itself against such attacks. As an alternative, the BadCSSChars setting can be modified if only one or two characters need to be removed from the banned character list.

LogoffUri and Session Management

The SiteMinder LogoffUri value needs to be set to the CA Clarity PPM logout URI so that when a CA Clarity PPM user logs out, the SiteMinder session is also invalidated. This can be changed in the Agent Configuration settings.

```
        logoffUri = /niku/app?action=security.logoutAction
```



Modify Agent Configuration: *ClarityAgentSettings*

Edit Parameter

Name: LogoffUri

Encrypted: □

● Plain ○ Multi-value

Value /niku/app?action=secur

Figure 21: SiteMinder Agent Configuration for LogoffUri

Chapter 4: Proxy from Apache HTTP Server

Typical Apache 2.x Configuration

In order for SiteMinder SSO to completely protect the CA Clarity PPM application server instances, CA Clarity PPM itself must prevent any users from bypassing the SiteMinder Web agent.

A typical configuration for SiteMinder with CA Clarity PPM would be to have the SiteMinder Web Agent running in an Apache HTTP server on the same server that is running the CA Clarity PPM application instance. In order to prevent users from reaching the CA Clarity PPM instance directly, bypassing the SSO web agent, CA Clarity PPM should be bound to the loopback address (127.0.0.1) on a non-privileged port (such as 8080). To do this, go to the Application settings in the CSA (CA Clarity PPM System Administrator) and adjust the "app" instance ***HTTP Port*** and ***Bind Address*** entries.

The Apache HTTP Server should be bound to a public IP address and a privileged port (80 or 443 if SSL is used). Apache should also be configured either as a reverse proxy with mod_proxy or with the Apache Tomcat Connector to communicate with the CA Clarity PPM application instance.

The typical configuration, documented below, uses the mod_proxy method. In the majority of configurations, mod_proxy is sufficient. The only functionality provided by the Apache Tomcat Connector that mod_proxy does not offer is software load balancing. The configuration of the Apache Tomcat Connector for Apache HTTP Server is not provided in this document, but can be found here:

http://tomcat.apache.org/connectors-doc/webserver_howto/apache.html

The following diagram illustrates the typical SiteMinder/CAClarity PPM configuration with an Apache server.

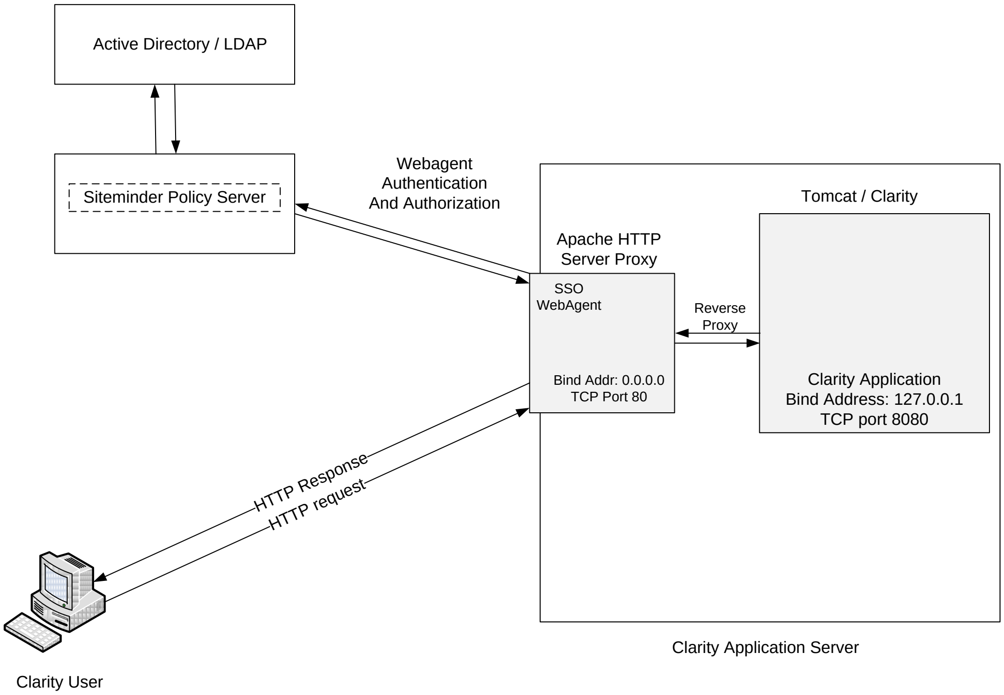


Figure 22: Typical SiteMinder/Apache/Clarity Environment

The following configuration can be used in an Apache 2.x web server to handle both the reverse proxy and the initial redirects required to land the end user on the CA Clarity PPM Overview page:

```
# Initial Clarity redirects for SSO
#
# NOTE: be sure to uncomment LoadLibrary lines for the following modules:
#
#      proxy_module
#      proxy_http_module
#      rewrite_module
#
RewriteEngine On
RewriteRule ^$  http://%{SERVER_NAME}/niku/app [R]
RewriteRule ^/$ http://%{SERVER_NAME}/niku/app [R]

ProxyPass /niku/ http://127.0.0.1:8080/niku/
ProxyPassReverse /niku/ http://127.0.0.1:8080/niku/
```

Figure 23: Apache 2.x Redirect/Proxy Configuration

Chapter 5: Proxy from Microsoft IIS

Microsoft IIS Configuration Overview

Microsoft IIS has an additional configuration requirement that is not found in Apache 2.x, because there is no built-in Proxy functionality in IIS. Instead, the Apache/Tomcat Connector must be used.

Note: When used with Microsoft IIS, the Apache Tomcat Connector automatically converts the underscore character in HTTP Headers to the dash character. Any SSO Header Token value that contains an underscore character will need to have the underscore replaced with a dash. In Chapters 2 and 3, the HTTP header token was set to **claritytoken** to prevent this issue from occurring.

Note: More information can be found on integrating the Apache Tomcat Connector with IIS at: Apache Tomcat Connector IIS How-to

The only major CA Clarity PPM configuration difference from the mod_proxy proxy method, presented in Chapter 4, is that the ***Tomcat Connector Bind Address*** and ***Tomcat Connector Port*** in the CSA Application settings must be set to the loopback address (127.0.0.1) and a non-privileged port (such as 30001) as described above in the reverse proxy Apache example.

In addition, for security purposes, the ***HTTP Port*** and ***Bind Address*** values should still be set to a non-privileged port (such as 8080) and the loopback address (127.0.0.1) respectively, to prevent end-users from bypassing the Web Agent in IIS.

The Apache Tomcat Connector will communicate directly with the ***Tomcat Connector Port*** in the CA Clarity PPM Application instance using the **ajp13** protocol (not HTTP).

The following diagram illustrates the typical SiteMinder/CAClarity PPM configuration with an IIS server.

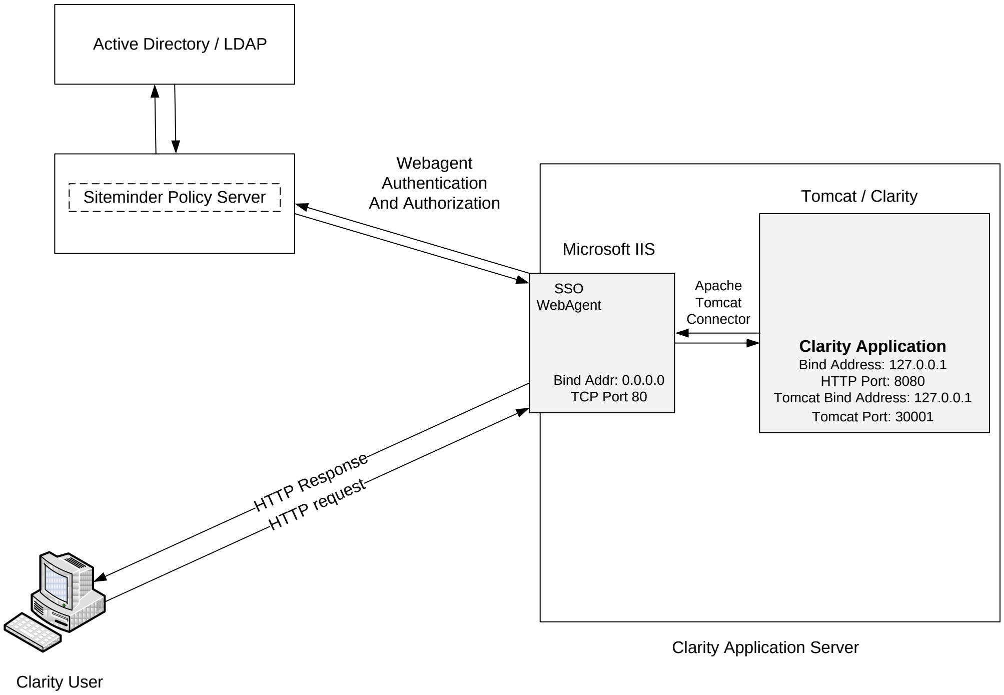


Figure 24: Typical SiteMinder/IIS/Clarity Environment

Installing the Apache Tomcat Connector in Microsoft IIS

Follow these seven steps to install the Apache Tomcat Connector in Microsoft IIS.

1: Download the ISAPI Redirector (isapi_redirect.dll)

The Apache Tomcat Connector for IIS has been implemented as an ISAPI filter DLL. This DLL can be downloaded from:

http://tomcat.apache.org/download-connectors.cgi

The latest version of the DLL, at the creation date of this document, is 1.2.28. The DLL needed for IIS can be downloaded from the win32/i386 "binaries" section and is named isapi_redirect-1.2.28.dll.

2: Place isapi_redirect.dll in the Tomcat bin directory

The Isapi Redirector DLL files include their version numbers when downloaded. You need to rename the file from "isapi_redirect-1.2.28.dll" to "isapi_redirect.dll" and place it in the "bin" directory of the Apache Tomcat installation (i.e. c:\niku\apache-tomcat-5.5.27\bin\isapi_redirect.dll).

Note: Rename the downloaded DLL to **isapi_redirect.dll**

3: Create the workers.properties and uriworkermap.properties files

The ***workers.properties*** and ***uriworkermap.properties*** files are read by the Isapi Redirector when IIS is started and provide configuration information necessary for the Tomcat Connector to reach the Tomcat application server.

The *workers.properties* file describes the hosts and ports used by the workers (Tomcat processes). The information in this file should match the bind address and port information from your CA Clarity PPM Tomcat configuration. Place this file into the "conf" directory under your Tomcat installation (c:\niku\apache-tomcat-5.5.27\conf\workers.properties).

```
# Define 1 real worker using ajp13
worker.list=worker1

# Set properties for worker1 (ajp13)
worker.worker1.type=ajp13
worker.worker1.host=127.0.0.1
worker.worker1.port=30001
```

Figure 25: workers.properties

The *uriworkermap.properties* file maps the URI patterns to workers. This file will contain one pattern, /niku/*. This file should also be placed into the "conf" directory under your Tomcat installation (c:\niku\apache-tomcat-5.5.27\conf\uriworkermap.properties).

```
# pattern for Clarity
/niku/*=worker1
```

Figure 26: uriworkermap.properties

4: Configure the ISAPI Redirector

In order to integrate the ISAPI redirector with IIS, several keys and values must be added to the Windows registry. Follow these steps:

1. Using "regedit", and being very careful to match the spelling exactly, create a new registry key named **HKEY_LOCAL_MACHINE\SOFTWARE\Apache Software Foundation\Jakarta Isapi Redirector\1.0**
2. Add a **String** value with the name **extension_uri** with a value of **/jakarta/isapi_redirect.dll**
3. Add a **String** value with the name **log_file** and a value pointing to the location you wish to have your connector logfile (for example: **c:\niku\apache-tomcat-5.5.27\logs\isapi_redirect.log**)
4. Add a **String** value with the name **log_level** and a value for your log level (valid values are **debug, info, error** or **emerg**). When finished with initial testing, be sure to set the level back to **error** to limit the amount of information logged.
5. Add a **String** value with the name **worker_file** and a value that is the full path to your *workers.properties* file (for example **c:\niku\apache-tomcat-5.5.27\conf\workers.properties**)

6. Add a **String** value with the name **worker_mount_file** and a value that is the full path to your *uriworkermap.properties* file (for example **c:\niku\apache-tomcat-5.5.27\conf\uriworkermap.properties**)

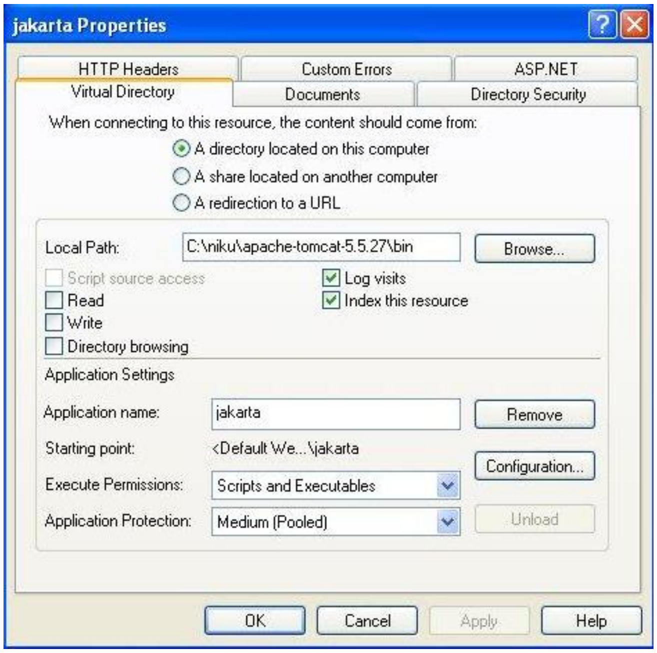
5: IIS Web Site Configuration

The steps for configuring IIS for v5/v6 are very different from those for v7. Below are two sections of configuration instructions specific to these versions. Follow the instructions for the version of IIS being used.

IIS v5/v6-Specific Configuration

Follow these configuration steps:

1. Using the IIS management console, add a new virtual directory to your IIS web site. The name of the virtual directory must be **jakarta**. Its physical path should be the directory where you placed the file **isapi_redirect.dll**. While creating this new virtual directory, assign it **execute** access.



When connecting to this resource, the content should come from:

● A directory located on this computer

○ A share located on another computer

○ A redirection to a URL

Local Path: C:\niku\apache-tomcat-5.5.27\bin

□ Script source access

□ Read

□ Write

□ Directory browsing

☑ Log visits

☑ Index this resource

Application name: jakarta

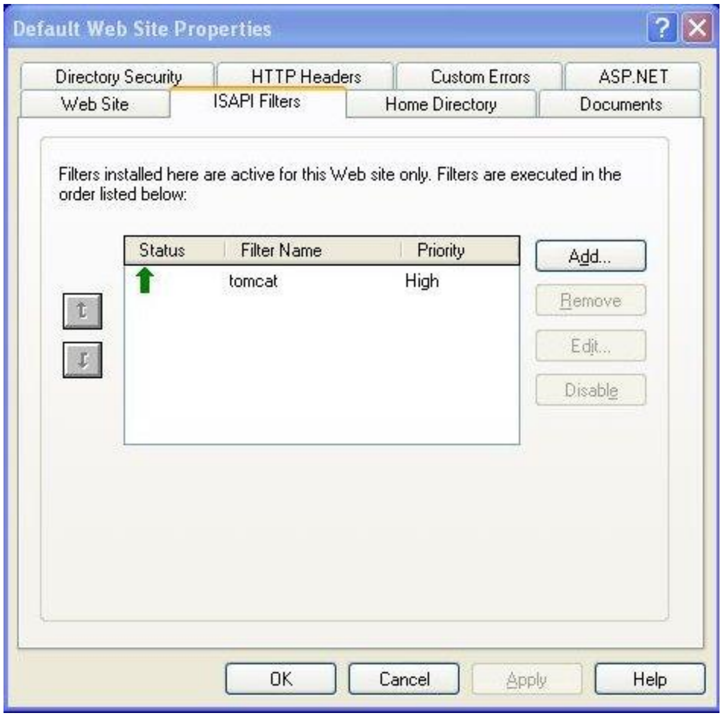
Starting point: <Default We...\jakarta

Execute Permissions: Scripts and Executables

Application Protection: Medium (Pooled)

Figure 27: Jakarta Virtual Directory

2. Using the IIS management console, add isapi_redirect.dll as a filter in your IIS web site. The name of the filter should reflect its task (for example, tomcat). Its executable must be the file *isapi_redirect.dll* you placed in the Tomcat "bin" directory earlier.



Filters installed here are active for this Web site only. Filters are executed in the order listed below:

Status	Filter Name	Priority
⬆	tomcat	High

Figure 28: Isapi Filters

3. If using IIS 6.0, add the Jakarta Isapi Redirector to the Web Service Extensions:

    a. Right-click on Web Service Extensions and choose "Add a new Web Service Extension".

    b. Enter "tomcat" for the Extension Name.

    c. Add the isapi_redirect.dll to the required files.

    d. Change the "Set Extension Status" value to Allowed.

    e. Click OK.

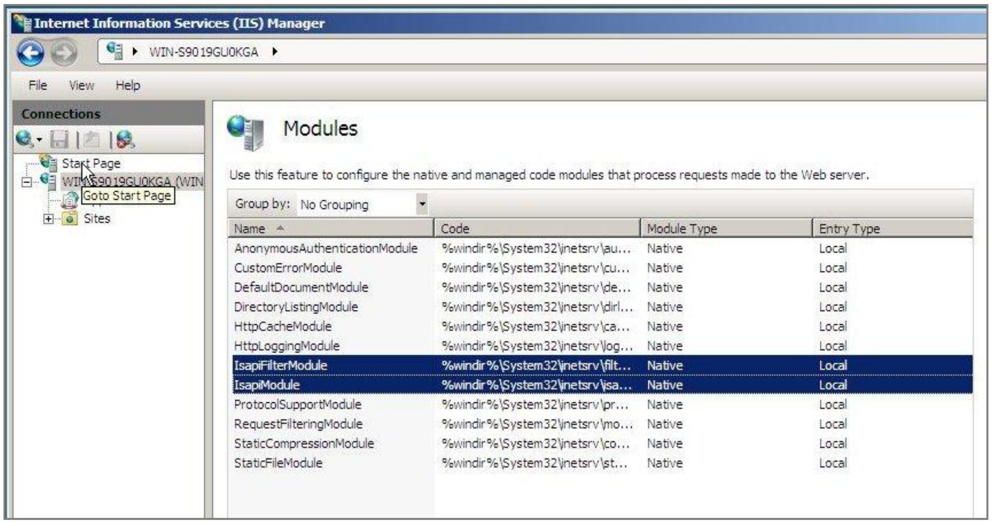
4. Restart IIS (stop and start the IIS Admin Service).

    Make sure that the tomcat filter is marked with a green up-pointing arrow. This may require a system reboot.

IIS 7-Specific Configuration

Follow these configuration steps:

1. Ensure that the IsapiFilterModule and IsapiModule are installed in IIS. These will not typically be installed by default. These modules can be installed using the Internet Information Services Manager.



Use this feature to configure the native and managed code modules that process requests made to the Web server.

Group by: No Grouping

Name	Code	Module Type	Entry Type
AnonymousAuthenticationModule	%windir%\System32\inetsrv\au...	Native	Local
CustomErrorModule	%windir%\System32\inetsrv\cu...	Native	Local
DefaultDocumentModule	%windir%\System32\inetsrv\de...	Native	Local
DirectoryListingModule	%windir%\System32\inetsrv\dirl...	Native	Local
HttpCacheModule	%windir%\System32\inetsrv\ca...	Native	Local
HttpLoggingModule	%windir%\System32\inetsrv\log...	Native	Local
IsapiFilterModule	%windir%\System32\inetsrv\filt...	Native	Local
IsapiModule	%windir%\System32\inetsrv\isa...	Native	Local
ProtocolSupportModule	%windir%\System32\inetsrv\pr...	Native	Local
RequestFilteringModule	%windir%\System32\inetsrv\mo...	Native	Local
StaticCompressionModule	%windir%\System32\inetsrv\co...	Native	Local
StaticFileModule	%windir%\System32\inetsrv\st...	Native	Local

Figure 29: IIS 7 Manager - Modules

2. Add the ISAPI Redirector as an ISAPI Filter to the IIS web site (Default Web Site is shown in the example). Follow these steps:

   a. Select the web site in the left navigation

   b. Double-click the “ISAPI Filters” icon in the window.

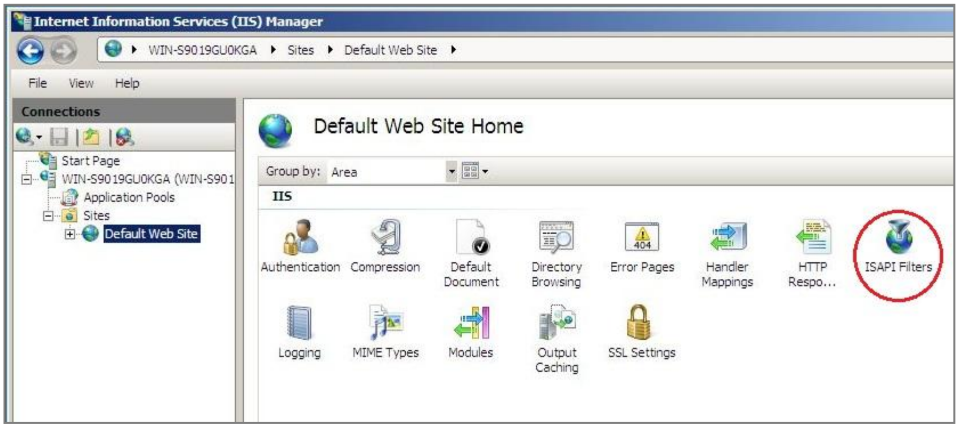
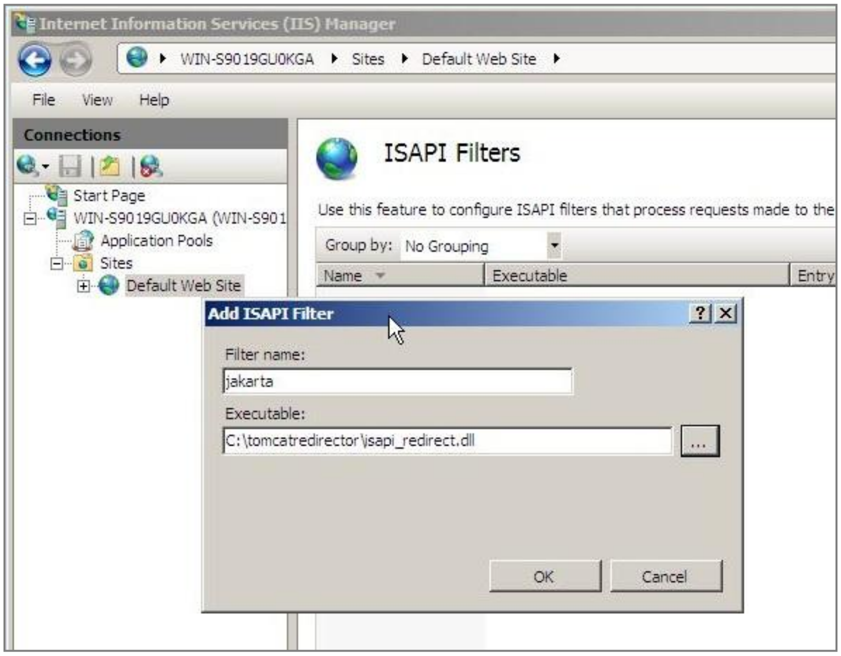


Figure 30: IIS 7 Manager – ISAPI Filters

c. Click Add in the Actions window pane.

The Filter Name must be **jakarta** and the Executable value should be the full path to the isapi_redirector.dll file.



Use this feature to configure ISAPI filters that process requests made to the

Group by: No Grouping

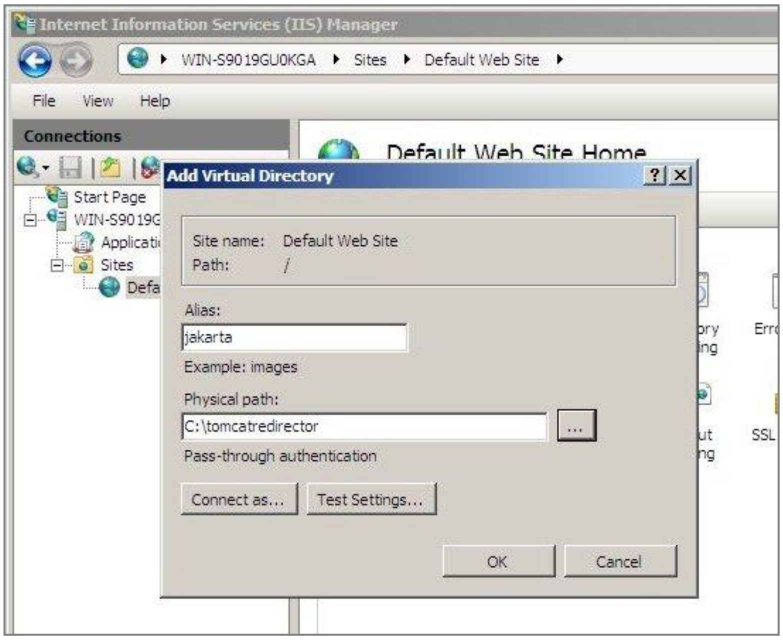
Filter name: jakarta

Executable: C:\tomcatredirector\isapi_redirect.dll

Figure 31: IIS 7 Manager – Add ISAPI Filter

3. Using the IIS Manager, right-click on your web site and select “Add Virtual Directory”.

   The name of the virtual directory must be **jakarta**. Its physical path should be the directory where you placed the file **isapi_redirect.dll**.



Site name: Default Web Site
Path: /

Alias: jakarta

Example: images

Physical path: C:\tomcatredirector

Pass-through authentication

Figure 32: IIS 7 Manager – Add Virtual Directory

4. After creating the new Virtual Directory, you must grant "execute" permissions to the ISAPI-dll Handler Mapping for the new directory. Follow these steps:

   a. Select the "Jakarta" virtual directory in the left navigation pane and then double-click the "Handler Mappings" icon in the main window.

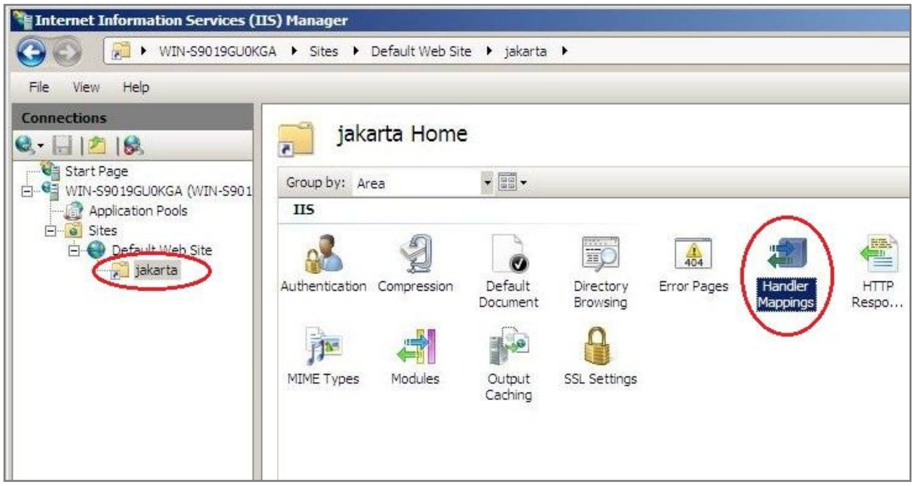
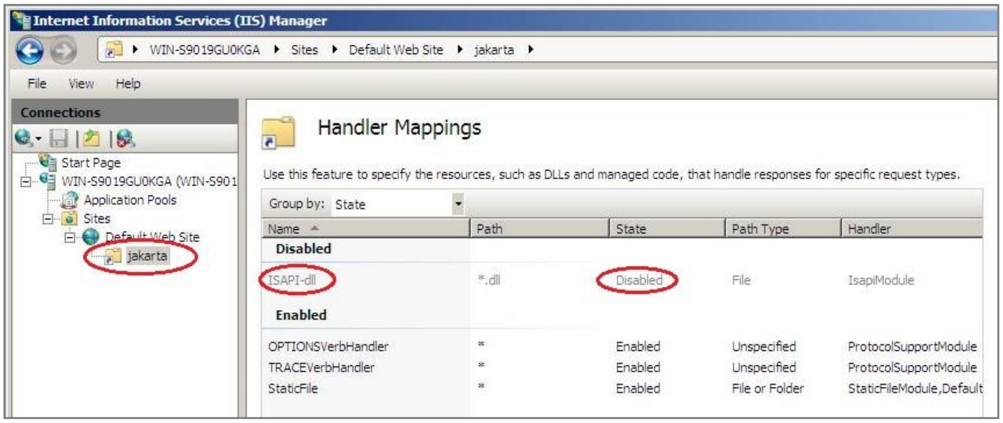


Figure 33: IIS 7 Manager – Handler Mappings

   b. Right-click on the disabled ISAPI-dll entry and select "Edit Feature Permissions."


Internet Information Services (IIS) Manager
WIN-S9019GU0KGA ▸ Sites ▸ Default Web Site ▸ jakarta ▸
File View Help
Connections
Start Page
WIN-S9019GU0KGA (WIN-S901
Application Pools
Sites
Default Web Site
jakarta

Handler Mappings

Use this feature to specify the resources, such as DLLs and managed code, that handle responses for specific request types.

Group by: State

Name	Path	State	Path Type	Handler
Disabled				
ISAPI-dll	*.dll	Disabled	File	IsapiModule
Enabled				
OPTIONSVerbHandler	*	Enabled	Unspecified	ProtocolSupportModule
TRACEVerbHandler	*	Enabled	Unspecified	ProtocolSupportModule
StaticFile	*	Enabled	File or Folder	StaticFileModule,Default

Figure 34: IIS 7 Manager – Disabled ISAPI-dll Handler Mapping

c. In the Edit Feature Permissions dialog, check the box next to "execute" and click OK.

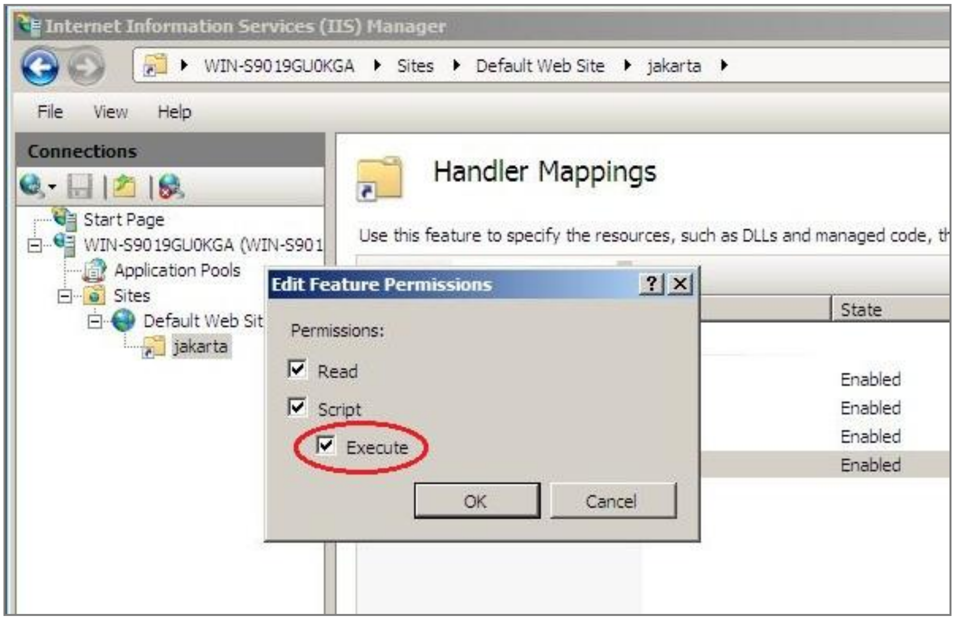

Internet Information Services (IIS) Manager
WIN-S9019GU0KGA ▸ Sites ▸ Default Web Site ▸ jakarta ▸
File View Help
Connections
Start Page
WIN-S9019GU0KGA (WIN-S901
Application Pools
Sites
Default Web Sit
jakarta
Handler Mappings
Use this feature to specify the resources, such as DLLs and managed code, th
Edit Feature Permissions
Permissions:
☑ Read
☑ Script
☑ Execute
OK
Cancel
State
Enabled
Enabled
Enabled
Enabled

Figure 35: IIS 7 Manager – Adding Execute Permission

5. Restart IIS Manager and test the ISAPI Redirector by opening the IIS website URL with a browser followed by /niku/app.

   For example: http://localhost/niku/app

6: Force CA SiteMinder Web Agent to Run as an ISAPI Filter in IIS v7

By default, the CA SiteMinder Web Agent runs as an ISAPI Filter in both IIS v5 and IIS v6. Under IIS v7 however, the CA Site Minder Web Agent runs as an IIS module. In order for the CA SiteMinder Web Agent to be executed *before* the ISAPI Redirector, we must force the CA SiteMinder Web Agent to run as an ISAPI Filter. This is done by means of an undocumented registry key in the CA SiteMinder configuration. Follow these steps:

1. Using RegEdit, navigate to the following registry entry for CA SiteMinder:

   ```
   HKEY_LOCAL_MACHINE\SOFTWARE\Netegrity\SiteMinder Web Agent\Microsoft IIS
   ```

2. Underneath this entry, add a new DWORD value called **IsapiMode** and set it to the value **1**.

3. Restart the IIS Admin Service so that the CA SiteMinder Web Agent is restarted with the new settings.

If these steps are not followed, the HTTP request will be proxied into Clarity *before* the CA SiteMinder We Agent runs and the user will not be authenticated for Single Sign On.

7: Configure the Root Context Re-direct in IIS

In order for IIS to redirect SSO users to the CA Clarity PPM overview landing page when they enter the web server at the root context, a redirect must be set up in IIS in the form of a default HTML document with a META-REFRESH directive to redirect the browser.

Follow these steps to create a default HTML document with META-REFRESH:

1. Create a document in the IIS web document root (typically c:\inetpub\wwwroot) called “default.htm”.
2. Enter the following content into that file:

   ```
   <html>
   <head>
   <meta http-equiv=”refresh” content=”0;url=/niku/app”/>
   </head>
   <body/>
   </html>
   ```

Chapter 6: Summary

Organizations can manage single sign-on access to CA Clarity PPM by integrating with CA SiteMinder. Real-time transactional security and integrated web services with CA SiteMinder rules enable security policies that evaluate dynamic data from a variety of local or external sources. These sources include web services and databases in real time. Cost and complexity are reduced by eliminating advanced security logic from web applications and centralizing it within CA SiteMinder policies.

Since the release of CA Clarity PPM 8.1.2, integration with CA SiteMinder for single sign-on is now available for all CA Clarity PPM components.

Note: For further information about CA Clarity PPM and CA SiteMinder, please refer to the *CA Clarity PPM Installation Guide*, the Siteminder Policy Server installation documentation, and the Siteminder Web Agent installation documentation.

It is highly recommended that CA Clarity PPM and/or CA SiteMinder specialists from CA Services be consulted as required when planning and/or implementing this integration. Please contact CA Services for assistance with your Clarity solution integration projects.

Index