NetIQ® SocialAccess

Installation and Configuration Guide

November 2015

Legal Notice

Contents

About this Book and the Library

The *Installation and Configuration Guide* for NetIQ® SocialAccess provides step-by-step instructions about how to install and configure the appliance.

Intended Audience

This book provides information for individuals responsible for implementing and administering the SocialAccess appliance. To use this book, you must also understand SAML 2.0.

Other Information in the Library

The library provides the following information resources:

Help

Provides context-sensitive information and step-by-step guidance for common tasks.

About NetIQ Corporation

We are a global, enterprise software company, with a focus on the three persistent challenges in your environment: Change, complexity and risk—and how we can help you control them.

Our Viewpoint

Adapting to change and managing complexity and risk are nothing new

In fact, of all the challenges you face, these are perhaps the most prominent variables that deny you the control you need to securely measure, monitor, and manage your physical, virtual, and cloud computing environments.

Enabling critical business services, better and faster

We believe that providing as much control as possible to IT organizations is the only way to enable timelier and cost effective delivery of services. Persistent pressures like change and complexity will only continue to increase as organizations continue to change and the technologies needed to manage them become inherently more complex.

Our Philosophy

Selling intelligent solutions, not just software

In order to provide reliable control, we first make sure we understand the real-world scenarios in which IT organizations like yours operate—day in and day out. That's the only way we can develop practical, intelligent IT solutions that successfully yield proven, measurable results. And that's so much more rewarding than simply selling software.

Driving your success is our passion

We place your success at the heart of how we do business. From product inception to deployment, we understand that you need IT solutions that work well and integrate seamlessly with your existing investments; you need ongoing support and training post-deployment; and you need someone that is truly easy to work with—for a change. Ultimately, when you succeed, we all succeed.

Our Solutions

- Identity & Access Governance
- Access Management
- Security Management
- Systems & Application Management
- Workload Management
- Service Management

Contacting Sales Support

For questions about products, pricing, and capabilities, contact your local partner. If you cannot contact your partner, contact our Sales Support team.

Worldwide:	www.netiq.com/about_netiq/officelocations.asp
United States and Canada:	1-888-323-6768
Email:	info@netiq.com
Website:	www.netiq.com

Contacting Technical Support

For specific product issues, contact our Technical Support team.

Worldwide:	www.netiq.com/support/contactinfo.asp
North and South America:	1-713-418-5555
Europe, Middle East, and Africa:	+353 (0) 91-782 677
Email:	support@netiq.com
Website:	www.netiq.com/support

Contacting Documentation Support

Our goal is to provide documentation that meets your needs. The documentation for this product is available on the NetIQ website in HTML and PDF formats on a page that does not require you to log in. If you have suggestions for documentation improvements, click **comment on this topic** at the bottom of any page in the HTML version of the documentation posted at www.netiq.com/documentation. You can also email Documentation-Feedback@netiq.com. We value your input and look forward to hearing from you.

Contacting the Online User Community

NetIQ Communities, the NetIQ online community, is a collaborative network connecting you to your peers and NetIQ experts. By providing more immediate information, useful links to helpful resources, and access to NetIQ experts, NetIQ Communities helps ensure you are mastering the knowledge you need to realize the full potential of IT investments upon which you rely. For more information, visit community.netiq.com.

1 Product Overview

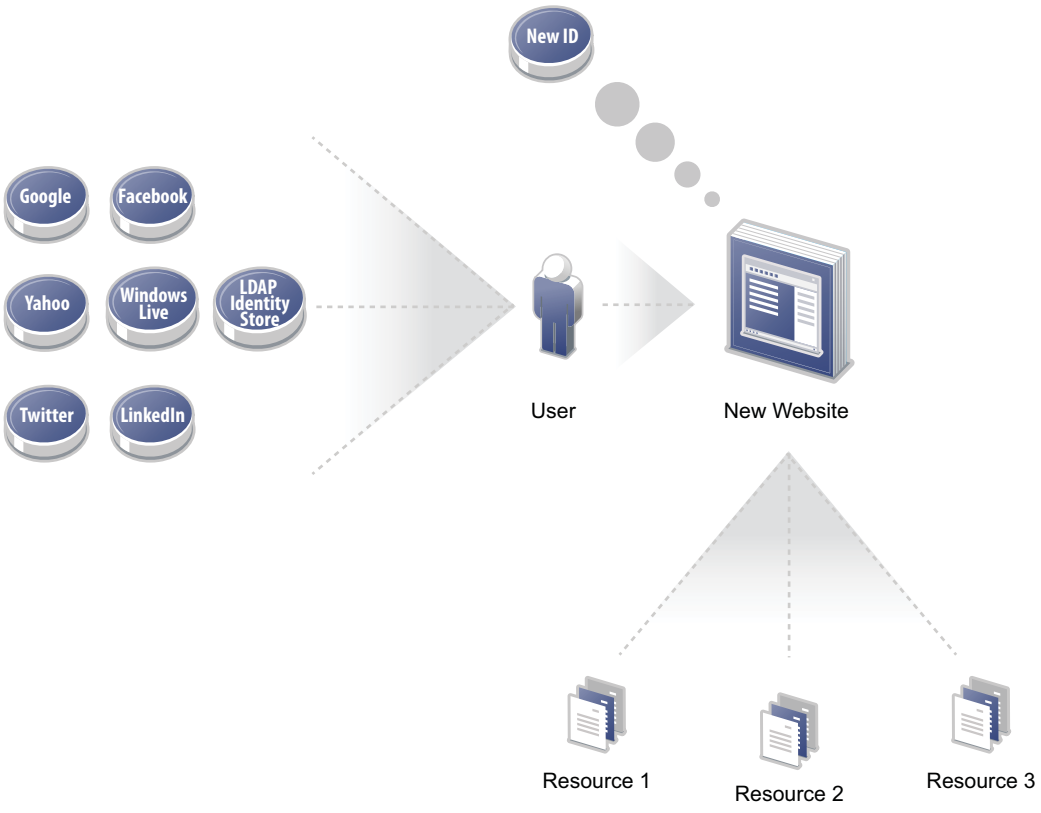
NetIQ SocialAccess is an appliance that simplifies a customer's experience accessing resources on the Internet.

1.1 Problems with Customers Accessing Resources on the Internet

Most businesses have an Internet presence and require customers to have an account to access the resources they provide on the Internet. As a customer this means you have multiple accounts and multiple passwords you must remember to access the resources you want on the Internet.

For example, the following graphic depicts the process a customer goes through to gain access to a new website they want to use.

Figure 1-1 *User Accesses a New Website*

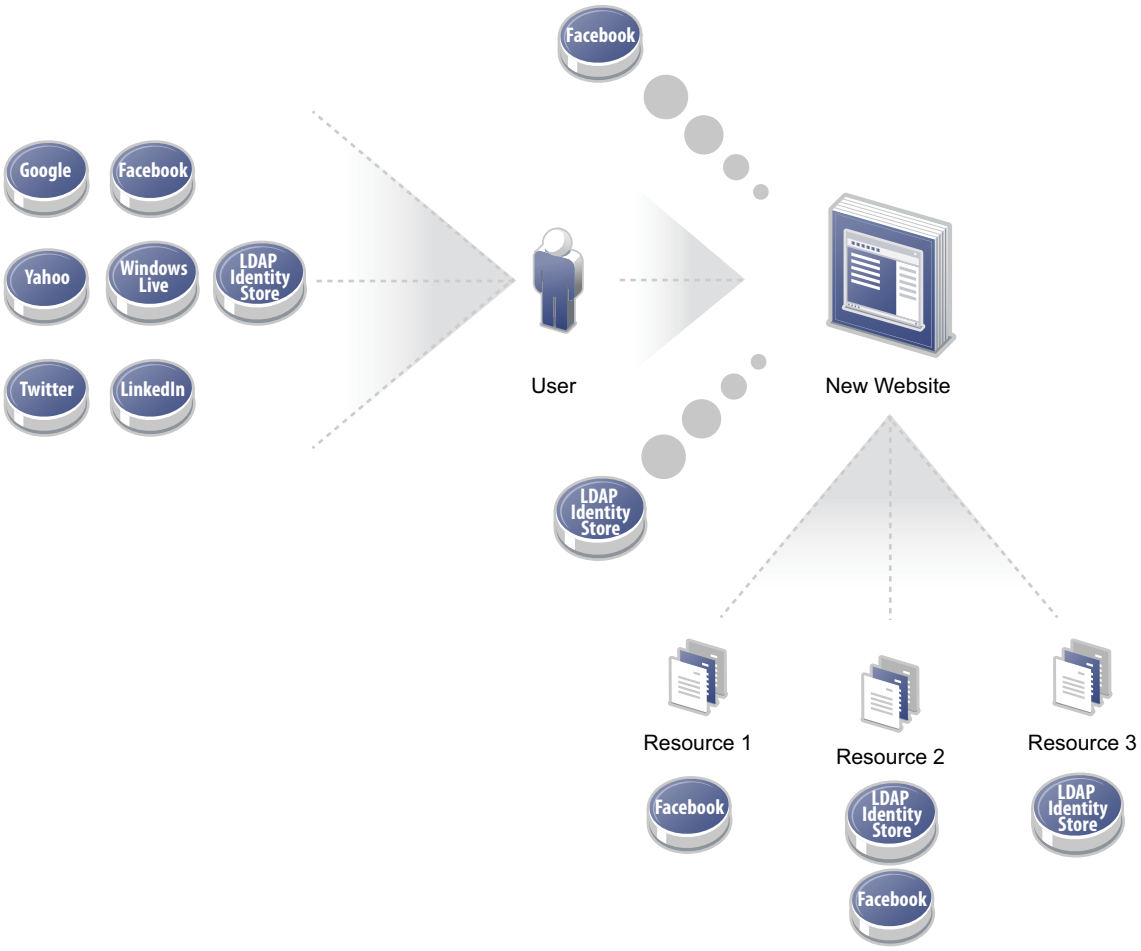


1. The customer already has multiple Internet account IDs and passwords.
2. The customer accesses the new website and the site requires the customer to create another account ID and password.
3. The customer creates a new account ID and password.
4. The customer accesses all of the resources available on the new website.

1.2 The Solution that SocialAccess Provides

SocialAccess provides a solution that allows customers to use existing Internet account IDs and passwords to gain access to a new website. The following graphic depicts the process a customer goes through using the SocialAccess solution.

Figure 1-2 *SocialAccess Solution*



1. The customer already has multiple Internet account IDs and passwords, but wants access to the new website.
2. The customer logs in to SocialAccess.

   SocialAccess displays a landing page with the different options the customer can use to authenticate.
3. The customer enters their existing Internet account ID and password for one of the sites listed. For example, the customer uses their Facebook account ID and password to log into the new website.
4. SocialAccess sends a request to Facebook to obtain customer attributes, which it then sends to the website.
5. SocialAccess grants the customer access to the new website, Resource 1, and Resource 2.
6. If the customer authenticated with the LDAP account ID, instead of Facebook, SocialAccess grants the customer access to the new website, Resource 2, and Resource 3.

SocialAccess controls which resources a customer can access depending on the type of existing account they use to access the new website.

IMPORTANT: SocialAccess does not attempt to log out users from any of the OAuth Providers. Authentication servers used by many OAuth Providers, such as Google, remember sessions and provide the default option for users to stay logged in. Unless users deselect the option to stay signed in, they are not prompted to enter their user name or password again. As a result, in environments with shared workstations, it is likely that a user could inherit another user's session. SocialAccess is intended to be used only for lightly secured resources, but does not guarantee that identities will be audited and corrected like a corporate directory.

1.3 How SocialAccess Works

SocialAccess allows you to configure multiple identity sources that customers can use to authenticate to your business' website. SocialAccess creates a SAML assertion using attributes obtained from the identity sources to allow SAML authentications into the resources associated with your website. For more information about configuring identity sources, see Chapter 4, "Configuring Identity Sources," on page 33.

For a sample network diagram, see Section 3.4.4, "A Sample Network Configuration," on page 22.

2 Installing SocialAccess

SocialAccess is a virtual appliance that you download and deploy into your IT environment.

2.1 Requirements

Use the information in the following table to verify that you meet the requirements for SocialAccess before deploying the appliance.

Table 2-1 *SocialAccess Requirements*

Components	Requirements
Supported Virtual Environments	The appliance requires one of the following virtual environments: ◆ VMware vSphere and vSphere Hypervisor 6.0 ◆ VMware vSphere and vSphere Hypervisor 5.5
Virtual System Guest Requirements	Minimum hardware requirements for each appliance node in the cluster: ◆ 60 GB disk space ◆ 2 cores ◆ 4 GB RAM **NOTE:** The default memory allocation is 8 GB, but you can change the memory allocation as needed before powering on the appliance. The appliance can be a heavy consumer of CPU, disk I/O, and network bandwidth. Performance can be adversely affected by other virtual machines with similar operational requirements deployed on the same VMware host server. As a best practice, ensure that you group or separate virtual machines on hosts and data stores to avoid resource conflicts for CPU, disk I/O, and network bandwidth. You can do this manually as you deploy virtual machines, or use affinity and anti-affinity rules if they are available in your VMware environment.
Client Workstations	**Administration:** Supported workstations for administration tasks: ◆ Microsoft Windows 10.*x*, 8.1, or 7.1 (no touch screens) ◆ Apple OS X (latest version) **Users:** Supported workstations for users: ◆ Microsoft Windows 10.*x*, 8.1, or 7.1 ◆ Apple OS X (latest version) ◆ Chromebooks (latest version)

Components	Requirements
Browsers	**Administration:** Supported browsers for administration tasks on a supported workstation: ◆ Mozilla Firefox (latest version) ◆ Google Chrome (latest version) ◆ Microsoft Internet Explorer 11 ◆ Apple Safari (latest version) **Users:** Supported browsers for users on a supported workstation: ◆ Mozilla Firefox (latest version) ◆ Google Chrome (latest version) ◆ Microsoft Internet Explorer 11 ◆ Apple Safari (latest version) **NOTE:** You must disable pop-up blockers to access the administration console. If you experience any issues with a supported browser, ensure that you have the latest version of the browser installed, or try another supported browser. Administering the appliance with Internet Explorer may be slower than with other supported browsers.
Cluster	Supported cluster configuration: ◆ The cluster can have up to five nodes. ◆ For optimal performance, each node should reside in the same IP subnet.
DNS	SocialAccess requires that all appliance nodes, administration workstations, end user workstations, and identity sources be able to resolve the public DNS name of the appliance.

2.2 Deploying the Appliance

The SocialAccess appliance is an Open Virtualization Format (OVF) virtual appliance. You must deploy the appliance to your VMware server.

The appliance must obtain an IP address through DHCP or have an assigned static IP address. NetIQ provides two different OVF files for each appliance, to accommodate DHCP and non-DHCP environments:

- **DHCP environment:** Use the `*.ovf` file for environments that have a DHCP server.
- **Non-DHCP environment:** Use the `*-vcenter.ovf` file for environments that do not have a DHCP server and need to use a static IP address.

To deploy the appliance in a VMware environment:

1. Download the appropriate file from the NetIQ Downloads web page (https://dl.netiq.com/).
2. (Conditional) If you are using Windows, extract the VMware image to access the available OVF file.
3. (Conditional) If you are using Linux, use the following command to extract the image:

   ```
   tar -zxvf vmware_image.tar.gz
   ```

4. (Conditional) If you have a DHCP server in your environment, deploy the `*.ovf` file to a specific ESXi host. For more information, see the VMware documentation.

5 (Conditional) If you do not have a DHCP server in your environment:

5a Deploy the `*-vcenter.ovf` file to a VMware vCenter Server, using either the command line tool “ovftool” or the VMware vSphere client.

5b Configure the appliance properties, ensuring that you change the `use_dhcp` property to false. Other required properties include the static IP address, subnet mask, default gateway, DNS server, and NTP server name.

TIP: If you deploy the appliance using the “ovftool,” you can configure the appliance properties from the command line and auto-start the VM so you do not have to use the vSphere client to configure the properties before starting the VM.

---

6 Power on the appliance, then proceed to Section 2.3, “Initializing the Appliance,” on page 17.

The initial boot configures the appliance. This process could take between five and ten minutes to complete. When the appliance is ready, it displays a welcome message with the initialization URL `https://appliance_ip_address/appliance/Init.html`.

2.3 Initializing the Appliance

Once you have deployed the appliance, you must initialize it.

1 Verify that you meet the requirements listed in Section 2.1, “Requirements,” on page 15.

2 From a supported browser, access the initialization web interface at the URL displayed on the appliance screen after it is deployed.

For example: `https://appliance_ip_address/appliance/Init.html`

IMPORTANT: This URL is case-sensitive, so ensure that you enter the non-variable portions of the URL exactly as illustrated in the example above.

---

3 Fill in the fields displayed to initialize the appliance.

4 Click **Finish**.

A successfully initialized appliance automatically redirects the browser to the administration console (`https://appliance_dns_name/appliance/index.html`).

5 Specify the appliance password, then proceed with Chapter 3, “Configuring the Appliance,” on page 19.

2.4 Re-initializing the Appliance

If you must re-initialize the appliance, use the following steps:

1 From a supported browser, access the initialization web interface:

`https://appliance_ip_address/appliance/Init.html`

2 Specify the password for the appliance, then change the configuration values as needed.

3 Click **Finish**.

3 Configuring the Appliance

After you have initialized the appliance, you must perform additional configuration for the appliance to work.

3.1 Accessing the Administration Console

After you properly initialize the appliance using the information in Section 2.3, “Initializing the Appliance,” on page 17, the browser automatically redirects to the administration console. If not, you can access the console as follows:

1. In a supported browser, specify `https://appliance_dns_name/appliance/index.html`.
2. Specify the appliance password you created during the initialization process.

3.2 Getting Started

Review the following checklist to help you get started using SocialAccess.

Table 3-1 *Getting Started*

❑	Steps	For more information, see...
❑	1. Register the appliance.	Section 3.3, “Registering SocialAccess,” on page 20
❑	2. Configure network options, such as adding a second NIC (network interface card).	Section 3.4, “Configuring Network Options,” on page 20
❑	3. Configure clustering.	Section 3.5, “Configuring Clustering,” on page 23
❑	4. (Optional) Configure event forwarding to Syslog.	Section 3.8, “Configuring SocialAccess to Forward Events to a Syslog Server,” on page 31
❑	5. (Optional) Configure Google Analytics to monitor usage.	Section 3.9, “Using Google Analytics as an External Dashboard,” on page 31
❑	6. Configure identity sources.	Chapter 4, “Configuring Identity Sources,” on page 33

3.3 Registering SocialAccess

SocialAccess provides a 30-day trial period. If you do not register the appliance within 30 days after installation, the appliance stops working. The bomb icon in the administration console displays how many days are left in the trial period.

For the purpose of meeting licensing requirements, when you register a single appliance, the cluster as a whole is considered to be registered. However, in order to use the NetIQ Customer Center (NCC) update channel to download and install software updates, you must register each node in the cluster separately. The bomb icon remains in the administration console if there are nodes in the cluster that have not yet been registered for channel updates. For more information about the update channel, see Section 12.4, “Updating the Appliance,” on page 82.

To register your appliance:

1. Log in to your Customer Center at https://www.netiq.com/customercenter (https://www.netiq.com/customercenter).

   The Customer Center is for NetIQ, Novell, and SUSE customers.
2. Click **My Products** > **Products**, then click **SocialAccess**.
3. Click the right arrow on the line next to the product to open a details page.
4. Select the **Activation code** value and copy it to the clipboard. You will need this code to register the appliance.
5. Log in to the appliance at `https://appliance_dns_name/appliance/index.html`.
6. Click the appliance node, then click **Register appliance**.
7. Enter the email address you used when you registered with the Customer Center.
8. Paste the Activation Code you copied to the clipboard from the Customer Center.
9. Click **Register**.
10. Repeat Step 6 through Step 9 for each node in the cluster.

When you have successfully registered all nodes in the cluster, the bomb icon disappears.

3.4 Configuring Network Options

SocialAccess contains a manual routing table, supports two NICs, and provides a forward proxy only for testing purposes.

3.4.1 Configuring the Forward Proxy

The forward proxy takes requests from the internal network and forwards these requests to the Internet.

NOTE: The forward proxy feature has the following limitations:

- The forward proxy is intended only for testing purposes, and is not supported in a production environment.

- When forward proxy is enabled, Simple Proxy connectors work only with web servers or resources (as specified in the **Connects to** field of the connector configuration) that are on the local segment. All simple proxied services must be reachable from the appliance without going through the forward proxy.

  In addition, before you configure a Simple Proxy connector to work with a web server on the local segment, you must add the IP address of the web server to the Forward Proxy's **Ignore List** configuration field. Otherwise, the simple proxy configuration will not be allowed, and the **Connects to** field will turn red. Wildcard entries in the **Ignore List** field are not supported.

To configure the forward proxy:

1. Access the administration console at `https://appliance_dns_name/appliance/index.html`, then log in with the password specified during the initialization process.
2. Drag and drop the **Forward Proxy** icon from the **Tools** palette to the **Tools** panel.
3. Use the following information to configure the forward proxy:

   Forward Proxy Server: Specify the IP address and port number for your proxy server.

   Ignore List: Specify any IP addresses with the associated DNS names that you want the forward proxy to ignore. For example, `127.0.0.0|localhost`.
4. Click **OK** to save your changes. Note that, as you click **OK**, the services restart and you must log in to the appliance again.

3.4.2 Configuring the Second NIC

SocialAccess allows you to configure two NICs for each node in the cluster. You can configure one NIC for the administrative network and a second NIC for the public network.

When you configure the second NIC, the SocialAccess appliance has only one global DNS name. In order for your users on the private network to access the correct network with the global DNS name for the appliance, you must do additional configuration on your network.

Two options allow users on the private network to access the SocialAccess appliance with the global DNS name:

- An entry in the local host file on each user's computer that resolves the global DNS name of the appliance to the private network
- A separate DNS server that routes all internal traffic to the global DNS name of the appliance

To configure the NICs:

1. Access the administration console at `https://appliance_dns_name/appliance/index.html`, then log in with the password specified during the initialization process.
2. Click a node icon, then click **Configure**.
3. To configure the first NIC, click the **Administration Interface** tab, then change the network settings for your administrative network.
4. Click **Apply** to save your changes.
5. To configure the second NIC, click the **Public Interface** tab.
6. Select **Enable Separate Public Interface**.
7. Configure the network settings for your public network.
8. Click **Apply** to save your changes.

9 Click **Close**.

10 Repeat Step 2 through Step 9 for each node in the cluster.

3.4.3 Configuring the Routing Table

SocialAccess provides a routing table for your use if your network has static routes. The routing table allows you to define the next hop in your network for the node in the cluster to reach the desired destination.

To configure the routing table for each node:

1 Access the administration console at `https://appliance_dns_name/appliance/index.html`, then log in with the password specified during the initialization process.

2 Click the node icon, then click **Configure**.

3 Click the **Routing** tab.

4 Specify the appropriate **Reverse Path Filter** setting. Reverse path filtering is used to prevent packets that arrived through one interface from leaving through a different interface. If in doubt, leave the default setting of **Strict mode**, since it prevents users from spoofing IP addresses from local subnets and reduces the likelihood of distributed denial-of-service (DDoS) attacks.

5 Click the plus sign (+) icon to add a route.

6 Define the desired route, then click **OK**.

7 (Optional) Add additional routes.

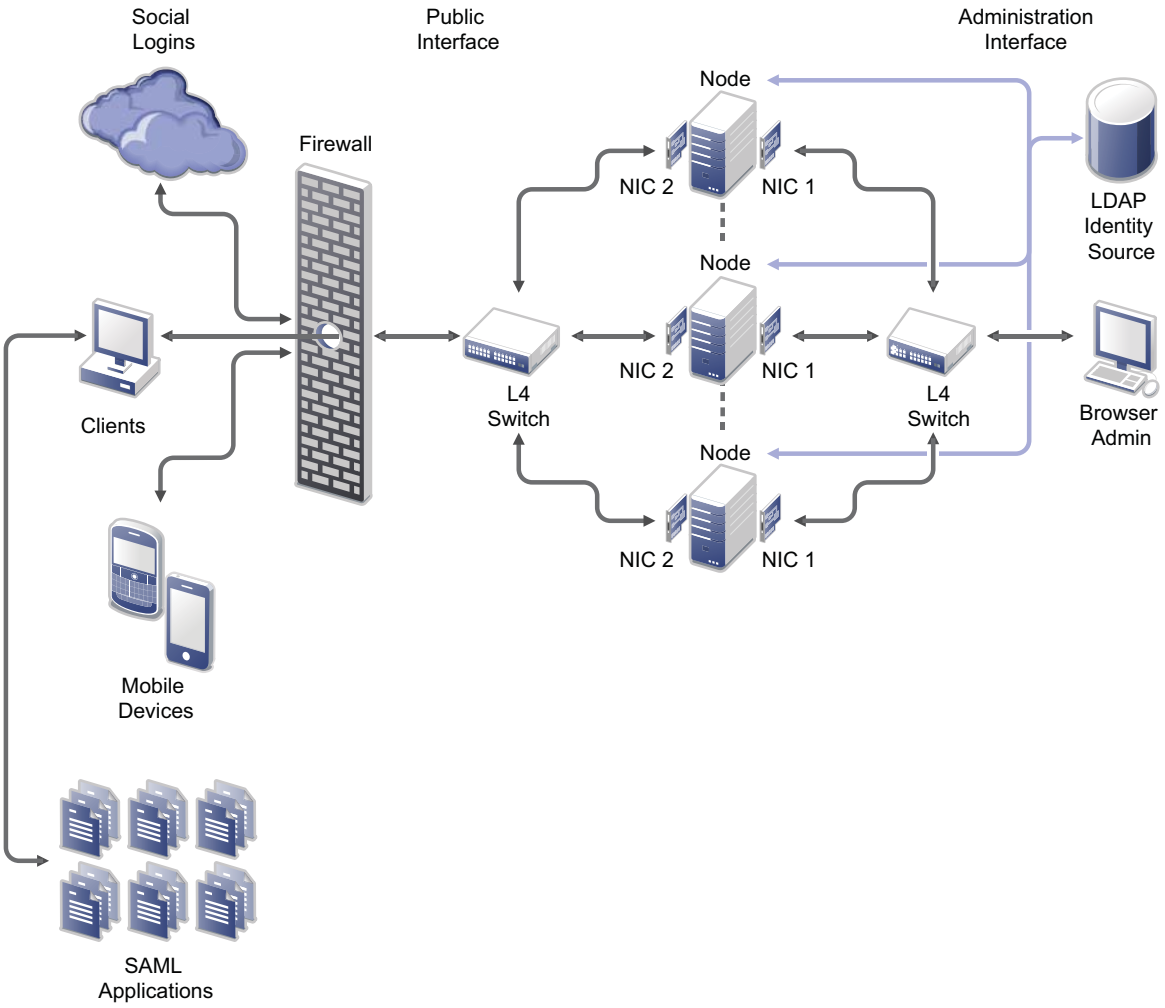
8 Click **Close**.

9 Repeat Step 2 through Step 8 for each node in the cluster.

3.4.4 A Sample Network Configuration

The following graphic depicts a possible network configuration using SocialAccess.

Figure 3-1 *Sample Network Diagram*



The network diagram shows that each node has both NICs enabled. The first NIC is the administration interface for the node and the second NIC is the public interface of the node. All of the administration and corporate information stays on the administration interface side of the network. All user requests and application requests communicate only on the public interface. This configuration provides a layer of security for your corporate information.

3.5 Configuring Clustering

You can cluster the SocialAccess appliance. By default, it is a single node cluster, but SocialAccess supports up to a five-node cluster. You add a node to the cluster by selecting **Join Cluster** during the initialization process.

3.5.1 Clustering Configuration Options

The Cluster icon configuration options are global settings for all nodes in the cluster. For more information about managing these configuration options, see the following sections:

- Section 12.3, “Changing Public DNS Name or NTP Server Settings, or Uploading New Certificates,” on page 81
- Section 11.2, “Configuring the Session Timeout for Users,” on page 80

3.5.2 Advantages of Clustering

Clustering in SocialAccess offers several advantages. Most of these advantages are available only if you configure an L4 switch or Round-robin DNS. The L4 switch is the best solution.

Disaster Recovery: Adding additional nodes to the cluster provides disaster recovery for your appliance. If one node stops running or becomes corrupt, you can promote another node to master.

High Availability for Authentications: SocialAccess provides high availability for authentications and the single sign-on service, when using an L4 switch in conjunction with clustering. This solution allows users to authenticate in case of problems with the nodes within the cluster. The L4 switch sends authentication requests to the nodes with which it can communicate.

Load Balancing: You can configure the L4 switch to distribute authentications to nodes so one node does not receive all authentication requests while other nodes sit idle.

Scalability: Configuring an L4 switch with clustering increases the scalability of SocialAccess. Each node in the cluster increases the number of possible simultaneous logins.

3.5.3 Managing Nodes in the Cluster

SocialAccess supports up to five nodes in a cluster. You add nodes to the cluster through the initialization process, and perform all other initialization tasks in the administration console.

Adding a Node to the Cluster

To add a node to the cluster:

1. Verify that the cluster is healthy.
   - All nodes must be running and communicating.
   - All components must be in a green state.
   - All failed nodes must be removed from the cluster.

   For more information about verifying that your cluster is healthy, see Section 13.3, “Troubleshooting Different States,” on page 86.
2. Download and deploy a new virtual machine (VM) for the new node.

   For more information, see Section 2.2, “Deploying the Appliance,” on page 16.
3. Initialize the appliance. Select **Join Cluster** as the first step to initialize the new node, then follow the on-screen prompts.

   For more information, see Section 2.3, “Initializing the Appliance,” on page 17.

   When initialization is complete, a login page appears.
4. Log in to the appliance and verify that the new appliance appears in the cluster. Wait until all spinner icons stop processing and all components are green before performing any other tasks.

The cluster is adding the node and several background processes are running. This final step could take up to an hour to complete.

5 Once the node is added to the cluster, register the node. For more information, see Section 3.3, “Registering SocialAccess,” on page 20.

Promoting a Node to Master

The first node that you install is the master node of the cluster by default. The master node runs provisioning, reporting, approvals, and policy mapping services. You can promote any node to become the master node.

To promote a node to master:

1 Verify that the cluster is healthy.

- All nodes must be running and communicating.
- All components must be in a green state.
- All failed nodes must be removed from the cluster.

For more information about verifying that your cluster is healthy, see Section 13.3, “Troubleshooting Different States,” on page 86.

2 Verify that all nodes in the cluster are running the same version of SocialAccess. If any nodes need to be updated, ensure that you update the nodes *before* you switch the master node. For more information, see Section 12.4, “Updating the Appliance,” on page 82.

3 Take a snapshot of the cluster.

4 In the administration console, click the node to become the master node, then click **Promote to master**.

An “M” appears on the front of the node icon indicating that it is now the master node. This process may take a while to complete. Watch for the node spinner icons to stop and health indicators to turn green before proceeding with any additional configuration changes.

The services move from the old master to the new master. The old master is now just a node in the cluster.

WARNING

- If the old master node is down when you promote another node to master, remove the old master from the cluster, then delete it from the VMware server. Otherwise, the appliance sees two master nodes and becomes corrupted.
- When you switch the master node, the logs and reports start again on the new master. If you do not have Syslog enabled, the historical logs and reporting data are lost. For more information, see Section 3.8, “Configuring SocialAccess to Forward Events to a Syslog Server,” on page 31.

Removing a Node from the Cluster

You can remove a node from the cluster if something is wrong with the node. However, ensure that you use the following steps to properly remove the node. If you simply delete a node from the cluster, the appliance deletes the node from the interface, but the VMware image still exists and continues to run. Leaving the VMware image running allows users to authenticate to a node that does not exist on the Admin page.

NOTE: After you remove a node, you cannot add the same VM instance back into the cluster. You must delete this instance of the appliance from your VMware server, then deploy another instance to the VMware server to add a node back into the cluster.

To remove a node from the cluster:

1. (Conditional) If the node you are removing is the master node, promote another node to be master before you remove the old node. For more information, see "Promoting a Node to Master" on page 25.
2. (Conditional) If you are using an L4 switch, delete the node from the L4 switch. For more information, see the L4 switch documentation.
3. In the administration console, click the node you want to remove from the cluster.
4. Click **Remove from cluster**.

   The administration console immediately shows that the node is gone, but it takes some time for the background processes to finish.
5. Stop the VMware image on the VMware server, and then delete the instance of the node from the VMware server.

3.5.4 Configuring an L4 Switch for Clustering

If you want high availability or load balancing, you must configure an L4 switch for the SocialAccess appliance. An L4 switch can be configured in many different ways. Use the following recommendations to configure the L4 switch to work with the appliance.

- **Heartbeat:** Use the following URL to define the heartbeat for the L4 switch:

  `https://appliance_ip_address/osp/h/heartbeat`

  The L4 switch uses the heartbeat to determine if the nodes in the cluster are running and working properly. The heartbeat URL returns a text message of Success and a 200 response code.
- **Persistence:** Also known as **sticky sessions**, persistence allows all subsequent requests from a client to be sent to the same node. To make this happen, select SSL session ID persistence when configuring the L4 switch.

Session persistence ensures that the same real server is used for the SocialAccess login and the subsequent application single sign-on. Using the same server allows caching for a series of related transactions, which can improve the server performance and reduce the latency of transactions. It removes the delay that might occur if the client sends a request to a new node instead of using the existing session to the same node. To ensure that transactions for the same client are forwarded to the same real server in a load-balanced cluster configuration:

- You can set the L4 switch to use IP-based persistence, which uses the user device's IP address to maintain an affinity between the user session and the same real server in the cluster. IP-based persistence fails if a user's device IP address changes between requests. It also fails if all user devices come through a proxy service where all transactions appear to come from the same IP address.
- You can set the L4 switch to use sticky-bit persistence. Sticky-bit persistence is problematic for L4 switches that do not support stickiness. Sticky sessions also do not work with browsers set to disable cookies.
- You can use a proxy approach for the identity provider nodes that does not depend on the L4 configuration. However, this solution can quickly become chatty.

3.6 Configuring Google reCAPTCHA

The Google reCAPTCHA tool helps protect your user login page against spam, malicious registrations, and other forms of attack where computers disguise themselves as humans. It provides an additional layer of security by displaying images of words that users must type in addition to their login credentials. Software bots typically cannot scan the images to provide a response.

Using reCAPTCHA helps prevent automated Denial of Service (DoS) attacks that can impact the performance of the appliance and the identity source. The tool uses the remote Google reCAPTCHA service to provide the images and verify the responses. If a response succeeds, the appliance verifies the user's authentication credentials against the identity source. If a response fails, the appliance fails the login attempt without processing the credentials, and re-displays the login page. Thus, the automated login attempts fail and cannot consume the processing resources of the appliance and identity source.

3.6.1 Requirements for reCAPTCHA

Ensure that your system meets the following requirements before you configure the Google reCAPTCHA tool:

- ❒ A SocialAccess appliance, installed and configured.
- ❒ One or more supported identity sources, with the connectors enabled and configured. The reCAPTCHA tool supports users from Active Directory and eDirectory identity sources.

  Each identity source should be configured with an intrusion detection policy. For more information, see Section 3.6.2, "Configuring Intrusion Detection for Failed Logins," on page 27.
- ❒ A Google reCAPTCHA account, configured on the Google reCAPTCHA website. For more information, see Section 3.6.3, "Configuring a Google reCAPTCHA Account," on page 28.

3.6.2 Configuring Intrusion Detection for Failed Logins

Someone who attempts to use more than a few unsuccessful passwords while trying to log on to your system might be a malicious user. reCAPTCHA cannot prevent attacks by anyone who can read the image. It cannot differentiate between malicious users and legitimate users. Using reCAPTCHA cannot prevent coordinated human DoS attacks. If users have unlimited attempts to enter their authentication credentials, reCAPTCHA also cannot help prevent attacks to find passwords.

To help limit the effectiveness of brute force or human attacks that bypass the reCAPTCHA protection, you should enable the user's identity source to respond to this type of potential attack by disabling the account for a preset period of time after a specified number of failed logon attempts.

The supported identity sources have the following built-in intrusion detection systems:

- **Active Directory Account Lockout Policy:** Active Directory allows you to specify an account lockout policy for users and global security groups in a domain. Set the policy on the domain group policy object from the domain controller.

  To configure the Account Lockout Policy settings:

  1. Log in as an Active Directory administrator user to the Windows Server that hosts Active Directory Domain Services (the domain controller).
  2. Configure the Account Lockout Policy on the group policy object for the domain controller.

     For more information, see the *Account Lockout Policy* (http://technet.microsoft.com/en-us/library/hh994563%28v=ws.10%29.aspx) in the Microsoft TechNet Library. (http://technet.microsoft.com/)

3. Verify that the **Account Lockout Threshold** value is higher than the number of failed login attempts you plan to specify for **Start reCAPTCHA at** in the reCAPTCHA tool.
4. Repeat these steps for each configured Active Directory identity source.

- **eDirectory Intruder Lockout Policy:** eDirectory allows you to enable Intruder Detection and specify an Intruder Lockout policy for the container object where your user objects reside.

  To configure the eDirectory Intruder Detection and Intruder Lockout Policy:

  1. Log in as the eDirectory administrator user to the management console for the eDirectory server.
  2. Configure Intruder Detection and the Intruder Lockout policy on the container object where your user objects reside.

     For more information, see "Setting Up Intruder Detection for All Users in a Container" (https://www.netiq.com/documentation/edir88/edir88/data/afxkmdi.html#a3p5g0i) in the *eDirectory 8.8 SP8 Administration Guide*.
  3. Verify that the Intruder Lockout value is higher than the number of failed login attempts you plan to specify for **Start reCAPTCHA at** in the reCAPTCHA tool.
  4. Repeat these steps for each configured eDirectory identity source.

After you have configured intrusion detection for the supported identity sources, continue with Section 3.6.3, "Configuring a Google reCAPTCHA Account," on page 28.

3.6.3 Configuring a Google reCAPTCHA Account

Before you configure the Google reCAPTCHA tool, you must configure an account to use for your domain at Google reCAPTCHA, and create a public and private key.

To configure a Google reCAPTCHA account to use for your appliance's domain:

1. Access the Google reCAPTCHA (https://www.google.com/recaptcha/) website.
2. Click **Get reCAPTCHA > Sign up Now**.
3. Log in using one of your Google accounts.

   For example, if you use your Gmail account, the reCAPTCHA account is associated with the Gmail account.
4. (Conditional) If this is not your first site, click **Add a New Site**. Otherwise, skip to the next step.
5. Specify a domain.

   Read the **Tips** for more information.
6. Click **Create** to add the domain.
7. Copy the **Public Key** and **Private Key** that the interface displays to use when you configure the identity source.
8. Continue with Section 3.6.4, "Configuring the reCAPTCHA Tool," on page 28.

3.6.4 Configuring the reCAPTCHA Tool

Before you configure the Google reCAPTCHA tool, ensure that you have set up intruder detection in the Active Directory and eDirectory identity sources, and created public and private keys for your appliance's domain at the Google reCAPTCHA website.

To configure the reCAPTCHA tool:

1. Using the identity source's native management tools, verify that their intrusion detection setup meets the requirements specified in "Configuring Intrusion Detection for Failed Logins" on page 27.
2. Log in with an appliance administrator account to the SocialAccess administration console at

   `https://appliance_dns_name/appliance/index.html`
3. In the **Identity Sources** panel, verify that you have configured an identity source for Active Directory or eDirectory, or both.
4. Drag and drop the reCAPTCHA tool from the **Tools** palette to the **Tools** panel.
5. Configure the reCAPTCHA tool as follows:

   Start reCAPTCHA at: Specify how many failed login attempts must occur before the login page displays the reCAPTCHA prompt. The value should be less than the lockout value set in the identity sources' intrusion detection system.

   - If the reCAPTCHA count is set to zero, the login page displays a reCAPTCHA prompt every time for all users. Every login requires user credentials and the reCAPTCHA response.
   - If the reCAPTCHA count is greater than zero, the login page displays the reCAPTCHA prompt only after the user login fails the specified number of times in the same browser window.

   Public Key: Paste the Public Key value from your reCAPTCHA account configuration for this appliance's domain.

   Private Key: Paste the Private Key value from your reCAPTCHA account configuration for this appliance's domain.

   For information about the public and private keys for your reCAPTCHA account, see Section 3.6.3, "Configuring a Google reCAPTCHA Account," on page 28.
6. Click **OK** to save the settings and enable the tool.
7. Click **Apply** to activate the configuration.
8. Wait while the service is activated across all nodes in the cluster. Do not attempt other configuration actions until the activation completes successfully.

3.7 Configuring the Authentication Filter to Set Session-Based Identity Information for a User

The SocialAccess single sign-on login is designed to authenticate a user against an identity source and to share this authentication with other protected applications. The authentication process does not provide extended functions to add, remove, or manage a user's identity information for the session. To address this need, SocialAccess provides the Authentication Filter tool.

The Authentication Filter integrates with the SocialAccess single sign-on process. After the user logs in, the filter intercepts the authentication process and sends the user's identity information from the identity source to your custom authentication scripts. You can add, remove, or set values for supported identity attributes. You can also set a cookie. You can interact with the user to gather input for those changes. After all of the encoded rules and associations are complete, SocialAccess stores the modified identity information in the session cache for the web services and applications.

The Authentication Filter tool is compatible with the ExtAPI library and the ExtUI library. It works with multiple scripting languages including PHP, Java, and Perl.

After you create your custom scripts, you must enable and configure the Authentication Filter tool in SocialAccess. The enabled filter automatically runs on each node in a SocialAccess cluster.

For information about creating custom authentication scripts to use with the Authentication Filter, see the *Authentication Filter Technical Reference for NetIQ CloudAccess* (https://www.netiq.com/documentation/cloudaccess/auth-filter_techref/data/auth-filter_techref.html). Although this Technical Reference was written for CloudAccess, the content and examples are equally applicable to SocialAccess.

Before you enable the Authentication Filter, set up your environment to meet the following requirements:

- A SocialAccess appliance, installed and configured.
- The Authentication Filter supports only applications that use session-based protocols. The filter stores the altered identity attributes and values in the session attribute cache.

  The Authentication Filter does not support applications that use sessionless protocols, because there is no session attribute cache to store the altered identity attributes and values. For example, the OAuth protocol is a sessionless protocol. Thus, the Authentication Filter does not support applications that use the OAuth Service Provider connector.
- On the ExtAPI server, create a script that uses the ExtAPI library commands to apply session-based authentication rules to an authenticated user's identity information. The Authentication Filter points to the URL for this file.
- If the session-based identity changes require user interaction:
  - On the ExtUI server, create a script that uses the ExtUI library commands to collect the user's session-based identity information, and return control to SocialAccess. The ExtAPI script should redirect the authentication session to the URL for this file.
  - On the ExtAPI server, create a redirect file configured with the ExtUI script's URL.

To enable the Authentication Filter:

1. Log in with an appliance administrator account to the administration console at `https://appliance_dns_name/appliance/index.html`.
2. Drag the **Authentication Filter** icon from the **Tools** palette and drop it in the **Tools** panel.
3. In the **Tools** panel, click the **Authentication Filter** icon, then click **Configure**.
4. In the Edit External Filter window, complete the following information:

   Display name: Specify a name for the filter. This name appears in the **Tools** panel of the console.

   Connects to: Specify the URL to the script that you want to run during the user SSO login.

   For example:

   `https://extapi_server_dns:port/path/extapi/index.php`

   Use HTTPS for secure SSL transfer of information. If you use an HTTP URL, information is not secure.

   Basic Auth User: (Optional) If login is required to access the URL, specify the user name to use in the basic authentication header.

   Basic Auth Password: (Conditional) If you specify a user name, specify the password for it.
5. Click **OK** to save and enable the filter settings.
6. Click **Apply** to activate the filter configuration.

7 Wait while the service is activated across all nodes in the cluster. Do not attempt other configuration actions until the activation completes successfully.

   In the **Appliances** panel, a green gear icon spins on top of each node until the activation is complete across all nodes in the cluster. In the **Tools** panel, a green status icon appears on the lower-left corner of the service icon. A yellow status icon appears if the URL uses HTTP instead of HTTPS because the traffic is not secure.

3.8 Configuring SocialAccess to Forward Events to a Syslog Server

You can configure SocialAccess to forward login and logout events to a syslog server.

To configure SocialAccess to forward events:

1 Access the administration console at `https://appliance_dns_name/appliance/index.html`, then log in with the password specified during the initialization process.
2 Drag and drop the Syslog tool from the **Tools** palette to the **Tools** panel.
3 In the **Tools** panel, click the Syslog tool, then click **Configure**.
4 Specify the IP address and the port of the syslog server.
5 Select the type of protocol to use: **UDP**, **TCP**, or **TLS**.
6 Click **OK** to save the tool settings.
7 Click **Apply** to activate event forwarding.

3.9 Using Google Analytics as an External Dashboard

SocialAccess enables administrators to use Google Analytics as an external dashboard to monitor and analyze SocialAccess usage. Once you have completed the free Google Analytics registration process for the SocialAccess appliance, data is available for analysis within a few hours. You can also do your own data mining with the API that Google provides. For more information, see the Google Analytics website (http://www.google.com/analytics/).

To set up Google Analytics for SocialAccess:

1 (Conditional) If you do not already have a Google account, set one up on the Google website.
2 Sign in to your Google account and select the option to register for Google Analytics.
3 Select the option to monitor a website and provide the base URL for the SocialAccess appliance. Google Analytics tracks both user and admin logins. For example, `https://appliance_dns`.
4 Specify an account name. This account name is only for managing Google Analytics and does not affect anything in SocialAccess. You can share this account name as needed.
5 Log in to the SocialAccess administration console.
6 Drag the Google Analytics tool from the **Tools** palette to the **Tools** panel.
7 Enter the Tracking ID (not the tracking code) that Google provided during the registration process and click **OK**.
8 Click **Apply** and wait for the appliance to update.

NOTE: If you have any issues with configuring the Google Analytics tool in the administration console, such as the tool being invisible on the Tools palette, verify that you do not have any adblockers running in your browser that may be interfering with administration tasks. You should be able to disable any adblockers on the web page itself.

4 Configuring Identity Sources

After you have configured the appliance, you need to configure the identity sources. SocialAccess uses identity sources for user authentication to SAML applications or of web services. You must configure the identity source and the connector for the identity source for the authentication process to work.

SocialAccess supports a number of identity sources and also provides two templates: OAuth 2.0 and OpenID Connect.

4.1 Configuring Active Directory as an Identity Source

SocialAccess supports Windows Server 2012 R2 and Windows Server 2008 R2.

To configure Active Directory as an identity source:

1. Verify that you have an Active Directory administrator account.
2. Log in to the SocialAccess administration console:

   `https://appliance_dns_name/appliance/index.html`
3. Drag and drop the connector for Active Directory from the **Identity Sources** palette to the **Identity Sources** panel.
4. Click the connector, then click **Configure**.
5. Use the following information to configure the connector for Active Directory:

   Credentials: Specify the fully distinguished LDAP format name and password of the Active Directory administrator account.

   Search Context: Specify the fully distinguished LDAP format of the context where the connector searches for user objects.

   Active Directory Servers: Specify the IP address and LDAP port of the Active Directory server where the user objects reside. Select **Enable LDAP SSL** to use port 636, or the default non-SSL port is 389.
6. (Optional) If you have custom attributes you want to map, click **Advanced Options**, then specify your custom attributes under **Attribute Mappings**.
7. Click **OK**, then click **Apply** to save the configuration.

The connector for Active Directory is now an identity source for user logins. Users and administrators can log in to SocialAccess using either their sAMAccountName or email address in Active Directory.

4.2 Configuring eDirectory as an Identity Source

SocialAccess supports eDirectory LDAP 8.8.8.

To configure eDirectory as an identity source:

1. Verify that you have an eDirectory administrator account with the following minimum rights for the connector for eDirectory to work:
   - **Property Rights**
     - **CN:** compare, read, inherit
     - **Description:** compare, read, inherit
     - **Given Name:** compare, read, inherit
     - **GUID:** compare, read, inherit
     - **Internet Email Address:** compare, read, inherit
     - **Login Disabled:** compare, read, inherit
     - **Member:** compare, read, inherit
     - **Member Of:** compare, read, inherit
     - **Surname:** compare, read, inherit
   - **Entry Rights:** browse, inherit
2. Log in to the SocialAccess administration console:

   `https://appliance_dns_name/appliance/index.html`
3. Drag and drop the connector for eDirectory from the **Identity Sources** palette to the **Identity Sources** panel.
4. Click the connector, then click **Configure**.
5. Use the following information to configure the connector for eDirectory:

   Credentials: Specify the fully distinguished LDAP format name and password of the eDirectory administrator account with the minimum rights.

   Search Context: Specify the fully distinguished LDAP format of the context where the connector searches for user objects.

   eDirectory Server: Specify the IP address and LDAP port of the eDirectory server that contains a Master or Read/Write replica of the partition where the user objects reside. Select **Enable LDAP SSL** to use port 636, or the default non-SSL port is 389.
6. (Optional) If you have custom attributes you want to map, click **Advanced Options**, then specify your custom attributes under **Attribute Mappings**.
7. Click **OK**, then click **Apply** to save the configuration.

The connector for eDirectory is now an identity source for user logins. Users and administrators can log in to SocialAccess using either their CN or email address in eDirectory.

4.3 Configuring reCAPTCHA for Active Directory or eDirectory

You can configure Google reCAPTCHA to work with Active Directory and eDirectory identity sources to provide an additional layer of security for the user login process. After a specified number of failed login attempts by users, reCAPTCHA displays a phrase that users must type in addition to their password. For more information about configuring reCAPTCHA, see Section 3.6, “Configuring Google reCAPTCHA,” on page 27.

4.4 Configuring Facebook as an Identity Source

To configure Facebook as an identity source:

1. Install the developer app to your profile from the Facebook Developers (https://developers.facebook.com) website.
2. On the developer site, click **Apps**, then click **Create a New App**.
3. Provide the following information:
   - Specify a **Display Name**.
   - From the **Category** list, select **Apps for Pages**.
4. Click **Create App**.
5. Click **Settings** in the left pane, then provide the following information:
   - In the **App Domains** field, create a new **App Domain** and specify your base DNS name (without `https`) in the format *`appliance_dns_name`*.
   - Specify a **Contact Email** address.
   - Click **+Add Platform**, then select **Website**.
   - Set the value of the **Site URL** field to your SocialAccess appliance publicly resolvable DNS name. For example, `https://appliance_dns_name`.
   - The other fields on this window are not required.
   - Click **Save Changes**.
6. Copy the **App ID** value and the **App Secret** value to use when you configure the connector for Facebook.
7. Log in to the SocialAccess administration console:

   `https://appliance_dns_name/appliance/index.html`
8. Drag and drop the connector for Facebook from the **Identity Sources** palette to the **Identity Sources** panel.
9. Click the connector, then click **Configure**.
10. Specify the **App ID** and **App Secret** values that you copied from the Facebook configuration.
11. Click **OK**, then click **Apply** to save the configuration.

The connector for Facebook is now an identity source for user logins.

4.5 Configuring Google as an Identity Source

To configure Google as an identity source:

1. Log in to Google Cloud Console (https://cloud.google.com/console) and create a new project.
2. Open the project you just created, then navigate to **API & Auth** > **Credentials**.
3. Click **Create New Client ID**.
4. Select **Web Application**.
5. Edit the **Authorized Javascript origin** to be the DNS name of your SocialAccess appliance.
6. Edit the **Redirect URI** to include the URL `https://dns_name/osp/a/t1/auth/oauth2/landingpad`, then change the *dns_name* to the DNS name of your SocialAccess appliance.
7. Click **Create Client ID**.

8 Copy the **Client ID** value and the **Client Secret** value to use when you configure the connector for Google.

9 Log in to the SocialAccess administration console:

`https://appliance_dns_name/appliance/index.html`

10 Drag and drop the connector for Google from the **Identity Sources** palette to the **Identity Sources** panel.

11 Click the connector, then click **Configure**.

12 Use the following information to configure the connector for Google:

Client ID: Specify the **Client ID** value from the Google configuration.

Client Secret: Specify the **Client Secret** value from the Google configuration.

13 Click **OK**, then click **Apply** to save the configuration.

The connector for Google is now an identity source for user logins.

4.6 Configuring LinkedIn as an Identity Source

Whether your SocialAccess application is an approved LinkedIn partner determines what user attributes SocialAccess is able to retrieve from LinkedIn. For more information about becoming a LinkedIn partner, see the LinkedIn website (https://developer.linkedin.com/partner-programs).

To configure LinkedIn as an identity source:

1 Log in to LinkedIn at the LinkedIn Developer website (https://www.linkedin.com/secure/developer).

2 Click **Add New Application**.

3 Create a new application with the following information:

OAuth Accept Redirect URL: Specify `https://dns_name/osp/a/t1/auth/saml2/sso` where the *dns_name* is the DNS name of the SocialAccess appliance.

JavaScript API Domains: Specify `https://dns_name` where the *dns_name* is the DNS name of the SocialAccess appliance.

4 Copy the **API Key** value and the **Secret Key** value to use when you configure the connector for LinkedIn.

5 Log in to the SocialAccess administration console:

`https://appliance_dns_name/appliance/index.html`

6 Drag and drop the connector for LinkedIn from the **Identity Sources** palette to the **Identity Sources** panel.

7 Click the connector, then click **Configure**.

8 Use the following information to configure the connector for LinkedIn:

API Key: Specify the **API Key** value from the LinkedIn configuration.

Secret Key: Specify the **Secret Key** value from the LinkedIn configuration.

LinkedIn Partner: Select this check box if your application has an approved partner relationship with LinkedIn. In addition to the user ID, name, email, and photo, SocialAccess can retrieve user phone, language, date of birth, and street address attributes for LinkedIn partners.

IMPORTANT: SocialAccess cannot validate your partner status. If you select this check box and your application is not an approved LinkedIn partner, authentication to LinkedIn will fail.

9 Click **OK**, then click **Apply** to save the configuration.

The connector for LinkedIn is now an identity source for user logins.

4.7 Configuring MSLive as an Identity Source

To configure MSLive as an identity source:

1 Create an application with a Windows Live login (MS Live) by following the instructions in the MSDN Library Getting Your Client ID for Web Authentication (http://msdn.microsoft.com/en-us/library/bb676626.aspx).

2 Edit the application, then add the DNS name of your SocialAccess appliance as the **Redirect domain** under the **API Settings**.

3 Copy the **Client ID** value and the **Client Secret** value to use when you configure the connector for MSLive.

4 Log in to the SocialAccess administration console:

`https://appliance_dns_name/appliance/index.html`

5 Drag and drop the connector for MSLive from the **Identity Sources** palette to the **Identity Sources** panel.

6 Click the connector, then click **Configure**.

7 Use the following information to configure the connector for MSLive:

Client ID: Specify the **Client ID** value from the MSLive configuration.

Client Secret ID: Specify the **Client Secret** value from the MSLive configuration.

8 Click **OK**, then click **Apply** to save the configuration.

The connector for MSLive is now an identity source for user logins.

4.8 Configuring Twitter as an Identity Source

To configure Twitter as an identity source:

1 Access the Twitter application at the Twitter Developer website (https://dev.twitter.com/apps).

2 Select **Create a new application**.

3 Create a new application with the following information:

Website: Specify the publicly resolvable DNS name of your SocialAccess appliance.

callbackURL: Specify `https://dns_name/osp/a/t1/auth/saml2/sso` where the *dns_name* is the DNS name of the SocialAccess appliance.

4 Copy the **Consumer Key** value and the **Consumer Secret** value to use when you configure the connector for Twitter.

5 Log in to the SocialAccess administration console:

`https://appliance_dns_name/appliance/index.html`

6 Drag and drop the connector for Twitter from the **Identity Sources** palette to the **Identity Sources** panel.

7 Click the connector, then click **Configure**.

8 Use the following information to configure the connector for Twitter:

Consumer Key: Specify the **Consumer Key** value from the Twitter configuration.

Consumer Secret: Specify the **Consumer Secret** value from the Twitter configuration.

9 Click **OK**, then click **Apply** to save the configuration.

The connector for Twitter is now an identity source for user logins.

NOTE: Twitter does not allow access to user email addresses, so SocialAccess cannot provide this information to the connector for NetIQ Access Manager. Users who log in to SocialAccess with a Twitter user account and then click an appmark for an Access Manager resource will receive the following error: `An Identity Provider response was received that failed to authenticate this session`.

4.9 Configuring Yahoo as an Identity Source

To configure Yahoo as an identity source:

1 Log in to the Yahoo Developer website (https://developer.apps.yahoo.com/).

You must have a Yahoo developer account to log in to the website.

2 Create a new application with the following information:

Application Type: Select **Web-based**.

Home Page URL: Specify `https://`*`dns_name`*`/osp/a/t1/auth/saml2/sso` where the *dns_name* is the DNS name of the SocialAccess appliance.

Access Scopes: Select **This app requires access to private user data**.

Callback Domain: Specify `http://`*`dns_name`*, where the *dns_name* is the DNS name for the SocialAccess appliance.

Permission Scopes: Enable the appropriate **Social Directory (Profiles)** permission as follows:

- **Read/Write Public and Private** to get *all* attribute values (ID, UserName, FirstName, LastName, BirthDate, Email, Photo, Gender, Language, StreetAddress, City, State, ZipCode, Country, Phone).
- **Read Public** or **Read/Write Public** to get only the following attribute values: ID, UserName, BirthDate, Photo, Gender, and Language

3 Click **Save and Change Consumer Key**.

4 Click **Verify Domain**, then click **Verify** next to the new domain name.

5 Copy the **Verification filename** value to use when you configure the connector for Yahoo, then click **Verify Domain** again and close the window.

6 Copy the **Consumer Key** and **Consumer Secret** values to use when you configure the connector for Yahoo.

7 Log in to the SocialAccess administration console:

`https://`*`appliance_dns_name`*`/appliance/index.html`

8 Drag and drop the connector for Yahoo from the **Identity Sources** palette to the **Identity Sources** panel.

9 Click the connector, then click **Configure**.

10 Specify the **Consumer Key**, **Consumer Secret**, and **Verification filename** values that you obtained from the Yahoo configuration.

11 Click **OK**, then click **Apply** to save the configuration.

The connector for Yahoo is now an identity source for user logins.

4.10 Configuring OAuth2 Sites as Identity Sources

SocialAccess provides a generic OAuth2 template. The OAuth2 template allows you to configure OAuth2 sites as identity sources for SocialAccess. For more information about OAuth2, see the OAuth2 website (http://oauth.net/2/).

To use the OAuth2 template:

1 Create an OAuth2 application that represents the SocialAccess appliance on the developer site you want to use as an identity source.

Creating an application does not require any coding.

2 Copy the following information into a document as you create the OAuth2 application to use when configuring the OAuth2 template:

- Client ID
- Client Secret ID
- Authentication URL
- Token URL or access token
- Profile URL
- (Conditional) Profile header
- Scope separator

3 Log in to the SocialAccess administration console:

`https://appliance_dns_name/appliance/index.html`

4 Drag and drop the OAuth2 template from the **Identity Sources** palette to the **Identity Sources** panel.

5 Click the template, then click **Configure**.

6 Use the information gathered in Step 2 to create your own connector for OAuth2 following the on-screen prompts.

7 (Optional) You can upload a login card image that is specific to your OAuth2 application in the **Login card image** field. Users see this image when they log in to SocialAccess.

The image can be a `.png`, `.jpg`, or `.gif` file. The file size is 215px x 50px and the file must be under 1 MB in size.

8 Click **OK**, then click **Apply** to save and create the connector for OAuth2.

It is common with some OAuth sources for the Token URL or Profile URL to require the oauth_token variable instead of the expected accessToken variable. To fix this, add the following:

`URL:?oauth_token{$accessToken}`

For example: `https://api.foursquare.com/v2/users/self?oauth_token={$accessToken}`

4.11 Configuring OpenID Connect Sites as Identity Sources

SocialAccess provides a generic OpenID Connect template. The OpenID Connect template allows you to configure OpenID Connect sites as identity sources for SocialAccess. For example, PayPal is an OpenID Connect site.

To use the OpenID Connect template:

1. Create an OpenID Connect application that represents the SocialAccess appliance on the developer site you want to use as an identity source.

   Creating an application does not require any coding.

2. Copy the following information into a document as you create the OpenID Connect application to use when configuring the OpenID Connect template:
   - (Optional) Discovery URL
   - (Optional) Register URL
   - Client ID
   - Client Secret ID
   - Authentication URL
   - Token URL
   - Profile URL

3. Log in to the SocialAccess administration console:

   `https://appliance_dns_name/appliance/index.html`

4. Drag and drop the OpenID Connect template from the **Identity Sources** palette to the **Identity Sources** panel.

5. Click the template, then click **Configure**.

6. Use the information gathered in Step 2 to create your own connector for OpenID Connect following the on-screen prompts.

7. (Optional) You can upload a login card image that is specific to your OpenID Connect application in the **Card image** field. Users see this image when they log in to SocialAccess.

   The image can be a `.png`, `.jpg`, or `.gif` file. The file size is 215px x 50px and the file must be under 1 MB in size.

8. Click **OK**, then click **Apply** to save and create the connector for OpenID Connect.

It is common with some OpenID Connect sources for the Token URL or Profile URL to require the oauth_token variable instead of the expected accessToken variable. To fix this, add the following:

```
URL:?oauth_token{$accessToken}
```

For example: `https://api.foursquare.com/v2/users/self?oauth_token={$accessToken}`

5 Configuring Application Connectors

SocialAccess provides several application connectors that are included in the appliance. This chapter describes configuration tasks that are common to multiple connectors. For more information about configuring specific connectors, see the following chapters:

- Chapter 6, “Configuring the SAML 2.0 Connector for ADFS,” on page 47
- Chapter 7, “Configuring the Connector for NetIQ Access Manager,” on page 55
- Chapter 8, “Configuring the Connector for OAuth2 Resources,” on page 57
- Chapter 9, “Configuring the Connector for Simple Proxy,” on page 61

In addition to configuring embedded connectors, you can create custom connectors. For more information, see Chapter 10, “Creating Custom Connectors with the Access Connector Toolkit,” on page 67.

5.1 Requirements for Connectors

As you configure connectors, ensure that you meet these general setup requirements:

- ❒ Ensure that the **Display Name** for each configured instance of a connector is unique for the appliance. The name allows you to identify a configured instance of the connector on the Applications panel of the administration console.
- ❒ The **Federation Instructions** on a connector’s Configuration page provide the information that you will use to configure federation for SocialAccess on the service provider site. The information identifies where on the service provider’s site to find the federation configuration capability as well as the field values and other guidance that you need to complete the required information.

  When you configure the connector, the federation instructions automatically provide the following information about your appliance as the identity provider:

  - The URL for single sign-on

    `https://appliance_dns_name/osp/a/t1/auth/saml2/sso`
  - The URL for single logout

    `https://appliance_dns_name/osp/a/t1/auth/app/logout`
  - The URL for the identity provider’s entityID

    `https://appliance_dns_name/osp/a/t1/auth/saml2/metadata`
  - The X.509 signing certificate for the appliance

    The application uses the certificate to set the trust relationship with SocialAccess.

    NOTE: When you copy the appliance’s signing certificate, ensure that you include all leading and trailing hyphens in the certificate’s Begin and End tags.

5.2 Viewing Connectors for Applications

SocialAccess displays the connectors in the following locations of the administration console:

- **Applications palette:** Displays unconfigured connectors that ship with the appliance.
- **Applications panel:** Displays configured connectors for the applications that you want to make available to users.

5.3 Providing Access to Applications for Users

After you configure the connectors for applications, you must provide a way for users to access the applications. SocialAccess provides the following portal pages for users:

- **Login page:** The login page allows users to enter their corporate credentials for authentication, such as user name and password. SocialAccess authenticates a user against your identity sources.

  The login page also exposes the authentication extensions of the appliance, such as Google reCAPTCHA, if configured.

  The URI for the login page is the public DNS name of the appliance. Provide this URl to your users:

  `https://appliance_dns_name`

- **Landing page:** The landing page contains the appmarks (linked icons) for accessing the applications that a user is entitled to use. The landing page appears after a user enters valid credentials and responds successfully to any additional authentication prompts.

  This page displays appmarks for an application only after the related connector is configured properly and the user has been authenticated. When a user clicks an appmark, SocialAccess shares identity information about the user with the application in order to establish the user's session. For more information about appmarks, see Section 5.4, "Configuring Appmarks for Connectors," on page 42.

5.4 Configuring Appmarks for Connectors

Appmarks are essentially bookmarks for applications. After you configure a connector for an application, you configure one or more appmarks to enable users to access the application in different ways. After users log in to SocialAccess, they see the appmarks on the landing page that they are entitled to see, according to the application settings for public access.

You can configure appmarks for any SSO connector. You can even configure multiple appmarks for the same connector. You can copy an existing appmark to create a new one.

NOTE: Appmarks for SSO connectors have no access control associated with them. If users know how to get to a service, they can access the service. Appmarks just add convenience to the user experience.

5.4.1 Understanding Appmark Options

You configure appmarks on the Appmarks tab in the Configuration window for the connector. On the Appmarks tab next to the name of the appmark in the blue bar are several icons for renaming, copying, disabling, or deleting the appmark. Use the mouseover text to identify the icon you want to use. You can view and edit appmark configuration options by clicking the blue bar or the plus sign (+) icon. The following appmark options are available:

Reset

This button restores the Appmarks tab to the default settings for the connector. Consider using this option if you have configured custom connectors that are not working as expected. Click **OK** and apply the changes to the appliance to see the default appmark settings.

Desktop browser

Enables the appmark to be visible on the SocialAccess landing page.

Initiate login at

Specifies whether the URL of the appmark on the landing page is the identity provider-initiated type or the service provider-initiated type. This option is not available for the OAuth2 Resources connector because the OAuth2 Resources connector offers only service provider-initiated authentication. It does not have an IDP-initiated mode.

URL

The URL that is to be used for the appmark. There are some replacement values that you can use. For more information, see Section 5.4.4, “Using Appmark Variables,” on page 45.

Icon

The icon that represents the application on the landing page. You can use a different custom icon for each connector to improve their usability for users.

5.4.2 Configuring an Appmark

After you have configured a connector for an application, you can configure an appmark to simplify access to that application from the user’s landing page.

To configure an appmark:

1. Log in with an appliance administrator account to the administration console:

   `https://appliance_dns_name/appliance/index.html`

2. (Conditional) If you have not already configured the connector for the application, drag it from the **Applications** palette to the **Applications** panel, then provide the appropriate information on the **Configuration** tab. The required information varies depending on the connector.
3. (Conditional) If you have already configured the connector for the application, click the connector on the **Applications** panel and click **Configure**.
4. Click the **Appmarks** tab.
5. Click the plus (**+**) sign next to the default created appmark.

6 The **Desktop browser** check box is selected by default. Complete the following steps to configure the appmark:

   6a (Conditional) If it is applicable to the connector, select the appropriate option from the **Initiate login at** list.

   6b Leave the default value in the **URL** field.

   6c (Optional) If you want to provide your own icon for the appmark, click the **X** on the **Icon** line to delete the default icon. Then browse to and select a `.png` file to represent the application on the browser's landing page.

7 Click **OK**, then click **Apply**.

The appliance reconfigures with the new change. After this process has completed, users who enter the appliance URL are redirected to a login page. They enter their user name and password and are presented with a landing page containing the appmark icon that links to the application.

5.4.3 Creating Multiple Appmarks for an Application

Application connectors can have multiple appmarks. You can create a new appmark from scratch, or you can copy an existing appmark to save time, especially if you want to create several appmarks and just change one or two options on each one. This procedure assumes you have already configured the connector.

To create a new appmark for a connector:

1 Log in with an appliance administrator account to the administration console:

   `https://`*`appliance_dns_name`*`/appliance/index.html`

2 Click the configured connector on the **Applications** panel, then click **Configure**.

3 Click the **Appmarks** tab, then do one of the following:

   - Click **New**
   - Click the **Copy** icon next to the existing appmark name

4 (Conditional) If you are copying an existing appmark, the **Name** field is pre-populated with `COPY_${DisplayName}`. You have several options:

   - You can accept this default name. (However, note that "COPY_" will be part of the name.)
   - You can change the display name by manually editing the text.
   - You can edit the display name by selecting from available variables. Type ${ at the end of the field, then select a variable from the list. For more information about the available variables, see Section 5.4.4, "Using Appmark Variables," on page 45.

5 Complete the appropriate fields for the appmark. For more information about available options, see Section 5.4.1, "Understanding Appmark Options," on page 43.

6 Click **OK**, then click **Apply** to update the appliance.

5.4.4 Using Appmark Variables

Each connector has different configuration settings and variables, and some appmarks need to contain information from the connector configuration to be useful. When you configure a connector, the Appmarks tab is automatically populated with one or more default appmarks, depending on the connector. The default settings contain some variables in the URL field.

You can use the variables that are available for a connector in the **Name** and **URL** fields if they are of the string type and have a value provided. To insert a variable, type `${` to display the available variables. Use the mouse or press the up/down arrow keys to select a variable. When you press the down arrow key, an additional box shows the resolved value. Press the up arrow key to close the resolved variables box. Some variables may not be resolvable until after you apply your changes on the appliance.

6 Configuring the SAML 2.0 Connector for ADFS

The connector for Active Directory Federation Services (ADFS) provides federated single sign-on access to ADFS with SAML 2.0 through SocialAccess. It does not support provisioning. The connector allows SocialAccess to authenticate a user against an enterprise identity source and to share this authentication with ADFS in order to establish the user's session.

SocialAccess includes this connector with the appliance. The connector appears automatically on the Applications palette in the administration console. After you configure the connector, you must also configure ADFS to work with the connector.

6.1 Requirements

Verify that you meet the following requirements before you import the connector:

- ❒ An understanding of identity federation using the SAML 2.0 protocol.

  For more information about SAML, see the OASIS website (https://wiki.oasis-open.org/security/FrontPage).

- ❒ An ADFS 2.0 system, installed and configured.
- ❒ Administrator access to the ADFS system. An understanding of ADFS and its management tools are presumed.
- ❒ An ADFS user account for each user who wants to authenticate to ADFS through the SocialAccess single sign-on service. The connector for ADFS does not provision user accounts.
- ❒ The location in the ADFS administration console where you will configure the SAML 2.0 federation for SocialAccess.

  When you configure the connector, the **Federation Instructions** provide the information that you will need to set up the federation in ADFS for SocialAccess. This information includes the metadata; a signing certificate for the appliance; the field values to use; and other guidance.

- ❒ The metadata file from the ADFS 2.0 system.

  `https://adfsserver/FederationMetadata/2007-06/FederationMetadata.xml`

  You will need the following information from the metadata file:

  - **Assertion Consumer Service URL:** The value in the **AssertionConsumerService** field with the HTTP-POST binding.
  - **EntityID:** The value in the **entityID** field.
  - **Logout URL:** The value in the **SingleLogoutService Location** field with the HTTP-POST binding.

- ❒ (Optional) An X.509 signing certificate from ADFS is required to support single logout. Communications use SSL regardless of whether you provide this certificate.

6.2 Configuring the Connector

After you import the connector, you must configure it to work with your ADFS system.

1. Log in as an administrator to the SocialAccess administration console:

   `https://appliance_dns_name/appliance/index.html`

2. Drag and drop the SAML 2.0 connector for ADFS from the **Applications** palette to the **Applications** panel.
3. On the **Configuration** page, specify the configuration properties.

   Use the information from the ADFS metadata file. The signing certificate from ADFS is optional.
4. Under **Assertion Attribute Mappings**, map the SAML Assertion attributes to the appropriate attributes in your identity source.
5. Expand the **Federation Instructions**, then copy and paste the instructions into a text file to use during the ADFS configuration for single sign-on.

   NOTE: You must use a text editor that does not introduce hard returns or additional white space. For example, use Notepad instead of Wordpad.

   ---
6. Click the **Appmarks** tab, then review and edit the default settings for the appmark.

   For more information, see Section 5.4, “Configuring Appmarks for Connectors,” on page 42.
7. Click **OK** to save the configuration.
8. Click **Apply** to commit the changes to the appliance.
9. Wait until the configuration changes have been applied on each node of the cluster.
10. Log in to ADFS as the ADFS administrator, then configure the SAML 2.0 federation for SocialAccess in the ADFS administration console.

    Use the information from the **Federation Instructions** in Step 5 to complete the setup.

    NOTE: When you copy the appliance’s signing certificate, ensure that you include all leading and trailing hyphens in the certificate’s Begin and End tags.

    ---
11. After you complete the configuration, users can log in through SocialAccess to single sign-on to the ADFS system. The SocialAccess login page URL is:

    `https://appliance_dns_name`

12. (Conditional) To allow Service Provider-initiated login, you must specify the Name ID format on the ADFS side. To do this, run the following PowerShell command:

    ```
    Set-ADFSClaimsProviderTrust -TargetName Display Name from Claims
    Provider Trust -RequiredNameIdFormat
    urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress
    ```

13. (Optional) If you want users to connect to SharePoint, proceed to Section 6.4, “Connecting to SharePoint,” on page 49.

6.3 Troubleshooting Certificate Errors

If you use a self-signed certificate and certificate chain errors occur, run the following PowerShell command:

```
Set-ADFSClaimsProviderTrust -TargetName Display Name from above -
SigningCertificateRevocationCheck None
```

6.4 Connecting to SharePoint

With additional configuration, the SAML 2.0 connector for ADFS allows users to single sign-on through SocialAccess to SharePoint as well as ADFS.

This section describes how you can leverage the claims-based single sign-on capabilities of ADFS and SharePoint to set up a hub model of federation through ADFS. In this hub, SocialAccess has a trusted relationship with ADFS as the identity provider, and ADFS has a trusted relationship with SharePoint as a claims-based federation provider. SharePoint accepts the claims-based assertions, and allows users to access federated SharePoint web applications. Using roles for claim-based single sign-on makes it easier for SharePoint site administrators to map role and organization claims to SharePoint groups.

To set up the relationships, you define the roles in the connector for ADFS that ADFS and SharePoint will use for the claims-based single sign-on. The connector adds the role information to the identity information in assertions that it sends to ADFS.

In ADFS, you configure claims rules that look for the email address and role of users, and then transform them for use by SharePoint. ADFS applies rules to the assertions from SocialAccess to transform them into role claims that the SharePoint web applications understand, and sends the role claims to SharePoint.

In SharePoint, you configure its Person Picker to look for the roles in the assertions from ADFS. SharePoint validates the assertion information, stores the information in its token cache, and issues a session cookie for the user. By default, SharePoint sets the session lifetime to be the same as the SAML token lifetime. In ADFS, you can specify the web single sign-on lifetime that determines the lifetime of the session cookie. Typically, the cookie expires when the user closes the browser window.

To set up this claims-based single sign-on federation hub:

- The SocialAccess administrator must modify the definition for the connector for ADFS to add two new roles to use for claims-based single sign-on, and then import and configure the modified connector.
- The ADFS administrator must configure a connection between SharePoint and ADFS, and define the rules for passing identity and role information from SocialAccess to SharePoint.
- The SharePoint administrator must modify the SharePoint People Picker to look for the roles in incoming assertions.
- The SharePoint administrator can add users to a SharePoint group based on the users' roles.

6.4.1 Requirements

Verify that you meet the following requirements:

❒ A SocialAccess appliance, installed and configured.

- ❒ One server with the following components installed:
  - ❒ Windows Server 2008 (or later) with the latest updates.
  - ❒ Active Directory with the latest updates.
  - ❒ ADFS 2.0 with the latest updates.
- ❒ A SharePoint 2010 (or later) server with the latest updates, installed in the same domain as the ADFS server.
  - ❒ The SharePoint server should be connected to the ADFS server.

    For information about connecting the servers, see the following references in the Microsoft TechNet Library:

    - *How to Configure ADFS v 2.0 in SharePoint Server 2010* (http://technet.microsoft.com/en-us/library/hh305235%28v=office.14%29.aspx).
    - *Configure SAML-based Claims Authentication with ADFS in SharePoint 2013* (http://technet.microsoft.com/en-us/library/hh305235%28v=office.15%29.aspx)
  - ❒ Roles enabled within the SharePoint system using PowerShell scripts.

6.4.2 Adding Roles to the SAML 2.0 Connector for ADFS

You must modify the definitions in a SAML 2.0 connector for ADFS template file to add roles that will be used when ADFS sends role claims to SharePoint. These instructions create two roles: an administrator role called `ADMIN` and a user role called `USER`.

Modifying the SAML 2.0 Connector for ADFS Template

Use the NetIQ Access Connector Toolkit to modify the definitions in the connector for ADFS.

1. Obtain a copy of the ZIP file for the SAML 2.0 connector for ADFS.
2. Log in as a SocialAccess administrator to the Access Connector Toolkit at

   `https://`*`appliance_dns_name`*`/css/toolkit`
3. Click **Import**, browse to and select the connector's ZIP file, then click **OK**.
4. Click the **Display Name** link for the connector to open it in the Edit Connector Template window.
5. Click the **Assertions** tab, then on the left side of the screen, click the **Attributes** tab.
6. Click **New,** then create a new Role attribute to use for the SharePoint connection.

   6a Define the properties for the Role attribute:

   Name: Specify `http://schemas.microsoft.com/ws/2008/06/identity/claims/role`.

   Display Name: Specify `Role`.

   Encoding: Leave this field blank.

   Data Owner: Leave this field blank.

   Default Value: Leave this field blank.

   Required: Select **false** to make this attribute optional.

   Description: Specify `A role assigned to the user account.`

   Role Attribute: Select **true**, then continue to configure the role definitions.

   6b Under **Roles**, click **New,** specify the following information, then click **Save**.

   Name: Specify `ADMIN`.

Description: Specify `Administrator Role`.

6c Under **Roles**, click **New,** specify the following information, then click **Save**.

Name: Specify `USER`.

Description: Specify `User Role`.

6d Add or customize any additional roles that you need for the SharePoint environment, and save each one.

6e Click **Save** to save the Role attribute definition.

7 Click **Save** to apply the connector template changes.

8 Click the **Export** icon next to the **Display Name** for the connector template.

9 Save the ZIP file for use on this or another SocialAccess system.

10 Proceed to "Importing the Modified Connector" on page 51.

Importing the Modified Connector

After you modify the SAML 2.0 connector for ADFS, you must import the connector into SocialAccess.

1 Log in as an administrator to the SocialAccess administration console at

`https://appliance_dns_name/appliance/index.html`

2 On the Admin page, click the **Tools** icon on the toolbar, then click **Import connector template**.

3 Click **Browse**, then browse to and select the ZIP file for the modified SAML 2.0 connector for ADFS.

4 Click **Import**.

The Applications palette displays the modified SAML 2.0 connector for ADFS.

5 Proceed to "Configuring the Modified Connector" on page 51.

Configuring the Modified Connector

After you export and import the modified connector, you configure the connector by following the steps in Section 6.2, "Configuring the Connector," on page 48.

After you configure a SAML 2.0 connector for ADFS that supports SharePoint roles, you must modify ADFS and SharePoint to accept these roles. Proceed to "Modifying Claims Rules in the ADFS System" on page 51.

6.4.3 Modifying Claims Rules in the ADFS System

Before you begin, ensure that you have configured a connection between ADFS and SharePoint. In ADFS, you must define the claim rules for incoming assertions from SocialAccess and for outgoing assertions sent to SharePoint.

Adding Claims Rules for SharePoint Roles in Incoming Assertions

You must modify the ADFS claim rules between ADFS and SocialAccess. The purpose of these rules is to allow the user's email address and the role to pass through to SharePoint.

To add claim rules for incoming assertions from SocialAccess:

1. Log in to your ADFS system.
2. Access the **Claims Provider Trusts** for SocialAccess.
3. Click **Edit Claim Rules**.
4. Add two rules using the following information:
   - Rule 1
     - **Claim rule template:** Select **Pass Through or Filter an Incoming Claim**.
     - **Claim rule name:** Specify `pass nameID`.
     - **Incoming claim type:** Specify `Name ID`.
     - **Incoming name ID format:** Specify `Email`.
     - **Pass through all claim values:** Select this option.
   - Rule 2
     - **Claim rule template:** Select **Pass Through or Filter an Incoming Claim**.
     - **Claim rule name:** Specify `pass Roles`.
     - **Incoming claim type:** Specify `Roles`.
     - **Pass through all claim values:** Select this option.
5. Exit the Rule editor.
6. Proceed to "Adding Claims Rules for Transforming Assertions for SharePoint" on page 52.

Adding Claims Rules for Transforming Assertions for SharePoint

You must configure ADFS to map the user's Email Address to Login on the SharePoint system, and to send the user's role.

To add claim rules for assertions sent to SharePoint:

1. In the ADFS console, select **Trust Relationships** > **Relying Party Trusts**.
2. Right-click *Name of your SharePoint system*, then select **Edit Claim Rules**.
3. Add two rules with the following information:
   - Rule 1
     - **Claim rule template:** Select **Transform an Incoming Claim**.
     - **Claim rule name:** Specify `NameID to EmailAddress`.
     - **Incoming claim type:** Specify `Name ID`.
     - **Incoming name ID format:** Specify `Email`.
     - **Outgoing claim type:** Specify `E-mail Address`.
     - **Pass through all claim values:** Select this option.
   - Rule 2
     - **Claim rule template:** Select **Pass Through or Filter an Incoming Claim**.
     - **Claim rule name:** Specify `pass Roles`.
     - **Incoming claim type:** Specify `Roles`.
     - **Pass through all claim values:** Select this option.

4 Exit the Rule editor.

5 Proceed to Section 6.4.4, “Configuring the SharePoint People Picker to Use the Roles,” on page 53.

6.4.4 Configuring the SharePoint People Picker to Use the Roles

The default SharePoint People Picker configuration requires a repository of users and groups for the people picker to search. However, in a claims-based access model, the only information SharePoint has is the claims data associated with the current user’s SAML assertion.

Before you begin, ensure that you have roles enabled within the SharePoint system using PowerShell scripts.

After you complete the ADFS configuration, you must configure the SharePoint option of **People Picker** to use the roles ADMIN and USER for claims received from ADFS.

1 Where the SharePoint system grants access, select **People Picker**.

2 Under **ADFS**, select **Role**.

3 In the **Find** box, specify either `ADMIN` or `USER`.

This field must contain the name of the role you configure the connector to use in Section 6.4.2, “Adding Roles to the SAML 2.0 Connector for ADFS,” on page 50.

4 Select the role SharePoint returns, then assign the role to the group within SharePoint.

6.4.5 Troubleshooting SharePoint Issues

Use the following information if you encounter problems.

Issue: Error: The root of the certificate chain is not a trusted root authority.

Solution: You need to change the SharePoint server certificates. For detailed instructions, see *Root Certificate Chain not Trusted* (http://blogs.technet.com/b/speschka/archive/2010/02/13/root-of-certificate-chain-not-trusted-error-with-claims-authentication.aspx).

7 Configuring the Connector for NetIQ Access Manager

NetIQ has enhanced the integration between SocialAccess and NetIQ Access Manager 4.1.0 to simplify the configuration. If you are using an older version of Access Manager you must use the generic connector for NetIQ Access Manager. For more information, see Using NetIQ CloudAccess as a Trusted Identity Provider for NetIQ Access Manager (https://www.netiq.com/documentation/cloudaccess/nca-nam-integration_techref/data/nca-nam-integration_techref.html). Although this technical reference was written for CloudAccess, it is also applicable if you are using the generic connector for NetIQ Access Manager with SocialAccess.

The connector for NetIQ Access Manager configures your Access Manager Identity Server as a service provider (SP) that consumes the authentication information from SocialAccess. This allows the connector for NetIQ Access Manager to provide federated single sign-on access to Access Manager through SocialAccess. It does not support provisioning. The connector allows SocialAccess to authenticate a user against an identity source and to share this authentication with Access Manager in order to establish the user's session.

SocialAccess includes this connector with the appliance. The administration console displays the connector automatically once you have installed SocialAccess.

7.1 Requirements for the Connector for Access Manager

- ❒ A SocialAccess appliance, installed and configured.
- ❒ A NetIQ Access Manager 4.1.0.0-201 system, installed and configured.

  NOTE: You must have version 4.1.0.0-201 of NetIQ Access Manager installed. This connector does not work with any other version.

  Ensure that SSL communications are enabled for Identity Server and Access Gateway, and that both components are configured to trust the same signing certificate authority. For more information, see "Enabling SSL Communications" (https://www.netiq.com/documentation/netiqaccessmanager4/basicconfig/data/b6vcbhk.html) in the *NetIQ Access Manager Setup Guide* (https://www.netiq.com/documentation/netiqaccessmanager4/basicconfig/data/bookinfo.html).
- ❒ Access Manager user accounts for each user who wants the single sign-on service.
- ❒ If you use an eDirectory identity source for Access Manager and you need to provide access to Access Gateway protected resources that require a user name and password, you must enable Universal Password in eDirectory for the Access Manager LDAP connection.

NOTE: Universal Password Retrieval options must be properly set in the configuration of the Universal Password policy in eDirectory, so that it allows the password to be retrieved from the Access Manager user store.

For more information, see *Unable to retrieve Universal Password from eDirectory using PasswordFetchClass (TID 7007114)* (http://www.novell.com/support/kb/doc.php?id=7007114).

7.2 Configuring the Connector

You must configure the connector to work with your Access Manager system.

To configure the connector for Access Manager:

1. Access the administration console at `https://appliance_dns_name/appliance/index.html`, then log in with the password specified during the initialization process.
2. Drag the connector for NetIQ Access Manager from the **Applications** palette to the **Applications** panel.
3. Specify the following information to configure the connector:
   - A display name
   - The IP address or DNS name of the Access Manager administration console server
   - An administrator user name and password for Access Manager
   - The clustered IP address of the Identity Server (IDP)
4. Click **Advanced**.
   - **4a** Select the attribute that contains a user's name identifier in the SocialAccess identity source.
   - **4b** Select the matching attribute in the Access Manager identity source.
5. Click the **Appmarks** tab, then create an appmark for one or more of the Access Manager protected resource URLs listed.

   For more information, see Section 5.4, "Configuring Appmarks for Connectors," on page 42.
6. Click **OK** to save the configuration.
7. Click **Apply** to commit the changes to the appliance.

As long as you have met the requirements, the SocialAccess appliance creates the required components in Access Manager to make it a service provider.

7.3 Configuring Appmarks for Protected Resources in Access Manager

After you have configured the connector for Access Manager, you can configure appmarks for protected resources in Access Manager.

The default appmark for the connector for Access Manager uses the **Destination URL** field from the configuration. If you did not specify the **Destination URL**, you will end up at the Access Manager home page for the default appmark.

For more information, see Section 5.4, "Configuring Appmarks for Connectors," on page 42.

8 Configuring the Connector for OAuth2 Resources

The connector for OAuth 2 Resources provides simple authenticated access to a web service through SocialAccess. The connector allows SocialAccess to authenticate a user against your identity sources and to provide protected access to a destination web service.

SocialAccess includes this connector with the appliance. The connector appears automatically on the Applications palette in the administration console. After you configure the connector, you must also configure the OAuth2 client application.

The connector for OAuth2 Resources offers a simple authentication method as an alternative to federated single sign-on connectors that use SAML 2.0 or WS-Federation protocols. Protocols for federated access management provide a robust trust and security model that is an open standard and widely used. However, it does require the protocol's code to be installed on the protected services. Consider using the connector for OAuth2 Resources for smaller services that do not require the full security and trust that SAML or WS-Federation provides, and just need a simple method to validate and get identity information from a trusted source (the SocialAccess identity provider in this case).

By implementing the open standard OAuth 2.0 protocol, the connector for OAuth2 Resources behaves as an OAuth2 Authorization Server and Resource Server using the Authorization Code flow as detailed in the OAuth 2.0 Authorization Framework document at http://tools.ietf.org/html/rfc6749#section-4.1.

Using this connector, the SocialAccess appliance provides user authentication and all OAuth2 token creation and validation for access to a protected resource.

NOTE: The OAuth2 Resources connector provides SP-initiated authentication. It does not have an IDP-initiated mode.

8.1 Configuring the OAuth2 Client Application

When you configure the connector for OAuth2 Resources in SocialAccess, the Client ID, Client Secret, and OAuth Endpoint URLs are created automatically. This information must then be used to configure the OAuth2 client application. All configuration activities at the OAuth2 client application are out of band.

Enforcement of authorization or access control beyond the initial authentication and token creation process is the responsibility of the OAuth client application. For information about configuring the OAuth client application, refer to your OAuth client application documentation.

8.2 Configuring the Connector for OAuth2 Resources

You can configure instances of the OAuth2 Resources connector in one of the following ways:

- An instance of the connector per OAuth client application. This is the simplest method conceptually and matches how SAML connectors are used.
- Multiple OAuth client applications all configured within a single instance of the OAuth2 Resources connector. This means that all OAuth2 client applications would use the same schema (OpenID Connect or native), and would use the same Client ID and Client Secret. This configuration is simple to configure and maintain, but care should be taken to include only clients of the same trust level in a connector instance. Because all clients share the same client ID and secret, if one of the clients is compromised in any way, they are all compromised. Any of them could also masquerade as another client in some cases.

(Optional) For each OAuth client application, you can manually create appmarks so the SocialAccess landing page shows an icon for connection to the OAuth2 client application. Appmarks should be configured to point to the URL of the OAuth2 client application that will start the OAuth2 authentication process.

To configure the connector for OAuth2 Resources:

1. Log in as an administrator to the SocialAccess administration console:

   `https://appliance_dns_name/appliance/index.html`

2. Drag and drop the **OAuth Resources** connector from the **Applications** palette to the **Applications** panel.
3. On the **Configuration** tab, provide the following information:
   - **Display name**: Clearly identify the connector on the **Applications** panel of the administration console.
   - **Schema**: Specify whether the attributes that SocialAccess sends to the OAuth client follow OpenID Connect standard naming or use the Native schema names defined internally on the appliance.
   - **Allowed OAuth Client URI(s)**: Specify the whole path or just the host name for the OAuth2 client application. Using only the host name allows all paths on that domain. Since OAuth2 depends on SSL as one of its core security mechanisms, HTTPS should always be specified. For more information about configuring redirect URIs, see the following document: http://tools.ietf.org/html/rfc6749#section-10.6.
   - **OAuth Details (Client ID and Client Secret)**: Use this information to configure the OAuth2 client application.
   - **OAuth Endpoints (Auth URL, Token URL, and Profile URL)**: Use this information to configure the OAuth2 client application.
4. Click the **Appmarks** tab, then review and edit the default settings for the appmark.
5. Click **OK** to save the configuration.
6. Click **Apply** to commit the changes to the appliance.
7. Wait until the configuration changes have been applied on each node of the SocialAccess cluster.

After the OAuth2 Resources connector and OAuth client application have been configured, end users can access the protected resource by browsing to the URL of the OAuth client application (by entering the URL directly into the browser, using a bookmark or the landing page appmark, and so forth). If the user is not already authenticated to the SocialAccess appliance, the browser is redirected

to the SocialAccess login page and the user is prompted for login credentials. After a successful authentication or if the user is already authenticated to the appliance and is authorized to access the protected resource, the user gains access to the resource.

8.3 Supported OpenID Connect Schema

The OAuth Resources connector supports the OpenID Connect schema names listed in the following table.

Table 8-1 *OpenID Connect Schema*

Member	Type	Description
name	string	End user's full name in displayable form including all name parts, possibly including titles and suffixes, ordered according to the user's locale and preferences.
given_name	string	Given name(s) or first name(s) of the end user. Note that in some cultures, people can have multiple given names; all can be present, with the names being separated by space characters.
family_name	string	Surname(s) or last name(s) of the end user. Note that in some cultures, people can have multiple family names or no family name; all can be present, with the names being separated by space characters.
middle_name	string	Middle name(s) of the end user. Note that in some cultures, people can have multiple middle names; all can be present, with the names being separated by space characters. Also note that in some cultures, middle names are not used.
preferred_username	string	Shorthand name that the end user wishes to be referred to at the RP, such as janedoe or j.doe. This value *may* be any valid JSON string including special characters such as @, /, or whitespace. This value *must not* be relied upon to be unique by the RP. (See Section 2.5.3 (http://openid.net/specs/openid-connect-basic-1_0-28.html#claim.stability) of the OpenID Connect Basic Client Profile 1.0 document.)
picture	string	URL of the end user's profile picture. This URL *must* refer to an image file (for example, a PNG, JPEG, or GIF image file), rather than to a Web page containing an image. Note that this URL *should* specifically reference a profile photo of the end user suitable for displaying when describing the end user, rather than an arbitrary photo taken by the end user.
email	string	end user's preferred email address. Its value *must* conform to the RFC 5322 (http://openid.net/specs/openid-connect-basic-1_0-28.html#RFC5322) addr-spec syntax. This value *must not* be relied upon to be unique by the RP, as discussed in Section 2.5.3 (http://openid.net/specs/openid-connect-basic-1_0-28.html#claim.stability) of the OpenID Connect Basic Client Profile 1.0 document.
gender	string	End user's gender. Values defined by this specification are female and male. Other values *may* be used when neither of the defined values is applicable.

Member	Type	Description
birthdate	string	End user's birthday, represented as an ISO 8601:2004 (http://openid.net/specs/openid-connect-basic-1_0-28.html#ISO8601-2004)[ISO8601-2004] YYYY-MM-DD format. The year *may* be 0000, indicating that it is omitted. To represent only the year, YYYY format is allowed. Note that depending on the underlying platform's date related function, providing just year can result in varying month and day, so the implementers need to take this factor into account to correctly process the dates.
locale	string	End user's locale, represented as a BCP47 (http://openid.net/specs/openid-connect-basic-1_0-28.html#RFC5646) [RFC5646] language tag. This is typically an ISO 639-1 Alpha-2 (http://openid.net/specs/openid-connect-basic-1_0-28.html#ISO3166-1) [ISO639 1] language code in lowercase and an ISO 3166-1 Alpha-2 (http://openid.net/specs/openid-connect-basic-1_0-28.html#ISO3166-1) [ISO3166 1] country code in uppercase, separated by a dash. For example, en-US or fr-CA. As a compatibility note, some implementations have used an underscore as the separator rather than a dash, for example, en_US; Implementations *may* choose to accept this locale syntax as well.
phone_number	string	End user's preferred telephone number. E.164 (http://openid.net/specs/openid-connect-basic-1_0-28.html#E.164) [E.164] is *recommended* as the format of this Claim, for example, +1 (425) 555-1212 or +56 (2) 687 2400. If the phone number contains an extension, it is *recommended* that the extension be represented using the RFC 3966 (http://openid.net/specs/openid-connect-basic-1_0-28.html#RFC3966) [RFC3966] extension syntax, for example, +1 (604) 555-1234;ext=5678.

9 Configuring the Connector for Simple Proxy

The connector for Simple Proxy provides reverse proxy access to your enterprise web service through SocialAccess. The connector allows SocialAccess to authenticate a user against your identity sources and to provide protected access to a destination web service. You can configure the connector to protect access to the document root of the web server, or to protect access only to a path within the document root of the web server.

Every web server is different. Two common simple proxy scenarios are:

- **Simple website:** If your web server provides a single web service, you can protect access to the entire site by creating a connector for Simple Proxy. The connector points to the document root of the web server.
- **Multiple-service website:** If your server provides multiple web services, you can create a separate connector for Simple Proxy for each destination web service. Each connector points to a different independent path within the document root of the web server.

If the web service requires user identity information to control access or content, you can configure the connector to inject the authenticated user's identity attributes in query strings and headers sent to the web service. However, the connector cannot be used to provide single sign-on for web services that require passwords for access. It does not support provisioning.

This connector is embedded with the SocialAccess appliance, and is located on the Applications palette in the administration console. Each cluster supports multiple connectors for Simple Proxy.

9.1 Requirements for Simple Proxy

The connector for Simple Proxy enables reverse proxy access to an enterprise web server behind your firewall. It can support web services that employ user identity information to control access or display if you enable the identity injection policies that insert an authenticated user's identity attributes in query strings or headers of requests it sends to the web server. For more information, see Section 9.2, "Viewing or Customizing the Attributes for Identity Injection," on page 62.

For each proxy web service, the web service's content should be self-contained in that path. If the service depends on files that reside in parallel paths on the web server, you can specify a path at a higher level in the document root's directory structure, or reorganize the site's contents as needed.

The connector for Simple Proxy does not support the following:

- **Protected resources that require a password:** This proxy solution cannot be used with protected web services or applications that require an LDAP password to be included in the identity injection. The appliance cannot send a user's password for a proxy application to the back end web service.

  If the web server needs the user's password, you must find a workaround. For example, you could specify a static string that is accepted for all users.
- **Site redirects:** This proxy solution does not support site redirects to locations outside the protected path. It cannot follow paths to alternate websites.

IMPORTANT: The Access Gateway for NetIQ Access Manager provides solutions for more complex reverse proxies that support password injection and redirects. For more information, see "Managing Reverse Proxies and Authentication" in the *NetIQ Access Manager Access Gateway Guide*.

Before you configure a connector for Simple Proxy, ensure that your environment meets the following requirements:

- ❒ A SocialAccess appliance, installed and configured.
- ❒ A web server, configured and running behind the corporate firewall. Ensure that you have configured the authentication procedures and identity injection policy for the web service.

  You need the following information:

  - The primary DNS name or IP address of the web server.
  - Alternative DNS names or IP addresses for the web server, if any.
  - The port number that the web server uses to listen for requests, such as 8080 (non-secure) or 8443 (secure SSL).
  - If the web server requires secure communications with HTTPS.

    If you use HTTPS, the value that you specify for the web server's DNS name or IP address in the connector must match the CN in the web server's SSL certificate.
- ❒ Determine which web services you need to protect for your web server, and which users require access to each one.

9.2 Viewing or Customizing the Attributes for Identity Injection

The connector for Simple Proxy can inject an authenticated user's identity attributes in query strings and headers of communications sent from the appliance to the destination web service. The web server might use this information to determine whether the user should have access to the resource. It can also use the identity information to customize content on the web page. For example, when a user whose first name is Joe (as specified in the identity source) navigates to the destination web page, he might see "Welcome: Joe" at the top of his browser window.

9.2.1 Understanding Identity Attributes

In the connector for Simple Proxy, you can enable or disable the following identity injection policies. Both policies are enabled by default.

- **Inject Identity in Query:** If you enable this option, when a user navigates to the connector's destination web service, the service receives all of the user's identity attributes in the query string.

  WARNING: Injecting attributes in the query string could exceed the maximum URL length of 2083 characters.

- **Inject Identity in Header:** If you enable this option, when a user navigates to the connector's destination web service, the service receives all of the user's identity attributes as custom headers.

If you enable an injection policy, the connector sends all of the user's identity attributes, even if the values are unavailable (empty). For some applications, this is still useful information and the web service can use it to make access or display decisions.

WARNING: If you use HTTP for communications between the connector and the web service, the injected identity attributes are available as clear text to network packet sniffers.

Although the proxy service runs behind the firewall, consider configuring the connector's web service URL with HTTPS to protect the communication stream to the web service. If you use HTTPS, the value that you specify for the web server's DNS name or IP address in the connector must match the CN in the web server's SSL certificate.

The attribute values in the query strings parameters or header parameters sent to the web server are based on the following options in the identity source user interface:

Identity Source Parameter	Query String Parameter	Header Parameter
ID	ID	X-ID
Email	Email	X-Email
User name	UserName	X-UserName
First name	FirstName	X-FirstName
Middle name	MiddleName	X-MiddleName
Last name	LastName	X-LastName
Full name	FullName	X-FullName
Preferred name	PreferredName	X-PreferredName
Generational qualifier	GenerationalQualifier	X-GenerationalQualifier
Gender	Gender	X-Gender
Phone	Phone	X-Phone
Birthdate	BirthDate	X-BirthDate
Street address	StreetAddress	X-StreetAddress
City	City	X-City
State	State	X-State
ZIP code	ZipCode	X-ZipCode
Country	Country	X-Country
Language	Language	X-Language
Identity Type	IdentityType	X-IdentityType
X-Custom1	XCustom1	X-XCustom1
X-Custom2	XCustom2	X-XCustom2
X-Custom3	XCustom3	X-XCustom3
X-Custom4	XCustom4	X-XCustom4
X-Custom5	XCustom5	X-XCustom5

The IdentityType parameter for query strings and headers indicates the type of identity source that the appliance uses to authenticate the user, such as Active Directory or eDirectory.

9.2.2 Viewing Identity Attribute Mappings to Identity Source Attributes

In your identity source, the identity attributes are mapped to identity source attributes. You can view mappings in your identity source connector. For example, in eDirectory, `ID` is mapped to the `guid` attribute, `User name` is mapped to the `cn`, and so on. You can change these mappings as needed for your environment, but any changes you make are global. You cannot change them on a per proxy or app basis.

To view identity attribute mappings in an identity source:

1. Log in as an administrator to the SocialAccess administration console:

   `https://`*`appliance_dns_name`*`/appliance/index.html`

2. In the **Identity Sources** panel, click the identity source, then click **Configure**.
3. Expand **Advanced Options**.
4. In the **Attribute Mappings** section, expand **Default** to view the list of the mappings of identity attributes to identity source attributes.
5. If you modify the settings, click **OK** to save your changes, then click **Apply**.

   Do not continue until the changes are applied to all nodes of the appliance cluster.
6. Repeat this process for each identity source that manages users who will access the destination web server.

9.2.3 Configuring Custom Identity Attributes

An identity injection sends all identity attributes. You cannot specify only a subset of attributes, add attributes, or remove attributes. However, you can map the `X-Custom`<1-5> attributes to attributes in your identity source. Ensure that you map the appropriate identity source attribute to each custom attribute across all of the identity sources for users who will access the destination web server.

To configure custom identity attributes in your identity source:

1. Log in as an administrator to the SocialAccess administration console:

   `https://`*`appliance_dns_name`*`/appliance/index.html`

2. In the **Identity Sources** panel, click the identity source, then click **Configure**.
3. Expand **Advanced Options**, then use the **Attribute Mappings** section to map custom attributes (`X-Custom`<1-5>) to attributes in your identity source.
4. Click **OK** to save your changes, then click **Apply**.

   Do not continue until the changes are applied to all nodes of the appliance cluster.
5. Repeat this setup for each identity source that manages users who will access the destination web server.

9.3 Configuring the Connector for Simple Proxy

Each connector for Simple Proxy can protect only a single web location. If the connector is set to protect the document root, then users can access all files served by the website. If the connector is set to protect a path under the document root, users can access only those files that reside in the path or its subdirectories.

To configure the connector for Simple Proxy:

1. Log in as an administrator to the SocialAccess administration console:

   `https://appliance_dns_name/appliance/index.html`

2. Drag and drop the connector for Simple Proxy from the **Applications** palette to the **Applications** panel.

3. On the **Configuration** tab, provide the following information:

   Display name: Specify the display name for the reverse proxy service. This name is also the default name of the appmark that appears on the user landing page.

   Local path: Specify a unique path on the appliance that will be used in the URL to associate traffic for the remote web service, such as `/myservice` or `servicexyz`. The path will be appended to the DNS name of the cluster for accessing the resource, and will be removed from the request before forwarding it to the web server.

   The local path must be unique across all connectors for Simple Proxy that you configure on the appliance. You can use alphanumeric characters a to z and 0 to 9, forward slashes (/), hyphens (-), and underscores (_). Spaces, uppercase characters, and other special characters are not supported. The path is not case sensitive. There is no length limit, but you should consider length restrictions for URLs and file system pathnames when you specify the character string.

   Connects to: Specify the URL of the destination web service that you want to protect.

   You can use HTTP (not secure) or HTTPS (secure) in the URL, depending on requirements of the web server. If you use HTTPS, the value that you specify for the DNS name or IP address must match the CN in the web server's SSL certificate. The connector automatically finds the SSL certificate and installs it for you if the URL uses HTTPS.

   You can specify the IP address or DNS name of the web server. Specify the port number if it is needed to access the location.

   Do one of the following:

   - Specify the root of the web server in order to protect all resources in the document root of the web server, including its subdirectories and their content.

     For example, you can specify the URL in any of the following formats:

     ```
     http://10.20.30.40
     http://10.20.30.40:8080
     https://myweb.example.com
     https://myweb.example.com:8443
     ```

   - Specify a path within the document root of the web server in order to protect only the resources in that path, including its subdirectories and their content through the **URL** field of the appmark. You must append the directory or query string to the end of the **URL** in the appmark.

     For example:

     Connects to: `http://www.youtube.com/watch`

     URL: `https://${PublicDNS}${PathFrag}?v=jAZveG_ptVU&sns=em`

     Similarly, for a file name of `http://151.155.160.14/examples/servlets/helloworld.html`, specify the following:

     Connects to: `http://151.155.160.14/examples/servlets`

     URL: `https://${PublicDNS}${PathFrag}/helloworld.html`

   Inject Identity in Query: Select this option to include the user identity attributes in the query strings that are sent to the **Connects to** URL. For more information, see Section 9.2, “Viewing or Customizing the Attributes for Identity Injection,” on page 62.

WARNING: Injecting attributes in the query string could exceed the maximum URL length of 2083 characters.

Inject Identity in Headers: Select this option to include the user identity attributes in the headers that are sent to the **Connects to** URL. For more information, see Section 9.2, "Viewing or Customizing the Attributes for Identity Injection," on page 62.

4 Expand **Advanced Options**, then configure the **Rewriter Options**.

The rewriter parses and searches the web content that passes through the appliance for URL references that qualify to be rewritten. URL references are rewritten when they meet the following conditions:

Strip Local path from query string: Enables URL references specified in the query strings to be rewritten with the published DNS name.

Strip Local path from POST data: Enables URL references specified in the post data to be rewritten with the published DNS name.

Strip Local path from REFERRER header: Enables URL references specified in the referrer headers to be rewritten with the published DNS name.

Alternative Host Names: URL references that match entries in this list are rewritten with the published DNS name. You can use any of the following formats. The entries are not case sensitive.

```
site.example.com
myhostname
10.10.2.10
http://<dns_name_or_ip_address>
http://<dns_name_or_ip_address>:port
https://<dns_name_or_ip_address>
https://<dns_name_or_ip_address>:port
```

You need to include names in this list if your web servers have any of the following configurations:

- If you have a cluster of web servers that are not sharing the same DNS name, you need to add their DNS names to this list.
- If your web server obtains content from another web server, the DNS name for this additional web server needs to be added to the list.
- If the web server listens on one port (for example, 80), and redirects the request to a secure port (for example, 443), the DNS name needs to be added to the list. This allows the response to be sent in the format that the user expects.
- If an application is written to use a private hostname, you need to add the private hostname to the list. For example, `http://<hostname>/index.html`.

5 Click the **Appmarks** tab, then review and edit the default settings for the appmark.

6 Click **OK** to save the configuration.

7 Click **Apply** to commit the changes to the appliance.

8 Wait for the configuration changes to be applied on each node of the SocialAccess cluster before performing other administration tasks in the console.

10 Creating Custom Connectors with the Access Connector Toolkit

SocialAccess provides the NetIQ Access Connector Toolkit (ACT) that allows you to create custom connectors. If you need help creating a custom connector to use with SocialAccess, Priority Support customers have the option to open a service request with NetIQ Technical Support (NTS) (http://www.netiq.com/support). NTS is available to provide toolkit support as well as to configure the connectors to work with integrated applications. Additional information from the SaaS provider is usually required.

NOTE: Before you contact NetIQ Technical Support, please complete the appropriate worksheet for the connector type that you want to create. See Appendix A, “Custom Connector Worksheets,” on page 91.

The Access Connector Toolkit facilitates custom connector development efforts without coding or scripting. You can create custom connectors for identity-aware web services or applications that use the following authentication methods for single sign-on:

- SAML 2.0
- WS-Federation
- SAML 2.0 Inbound (SAML-In)

NOTE: Although the Access Connector Toolkit displays options for creating custom connectors for web services or applications that use Basic SSO (forms-based) authentication for single sign-on, SocialAccess does not support Basic SSO custom connectors.

After you create a connector, you must export it from the toolkit as a file that you can import into SocialAccess. You can use the SocialAccess administration console to import and enable the connector, and to create appmarks for the web service or application.

10.1 Accessing the Access Connector Toolkit

The Access Connector Toolkit is a web application that you access through the SocialAccess appliance. Log in to the toolkit using SocialAccess administrator credentials at:

`https://appliance_dns_name/css/toolkit`

The Access Connector Toolkit does not currently provide a logout option, though the session does time out after 60 minutes of inactivity. Ensure that you close the browser after you finish working in the Access Connector Toolkit.

10.2 Toolkit Requirements

The Access Connector Toolkit is a web application that ships with SocialAccess. You can use the Access Connector Toolkit to create custom connectors if you have a SocialAccess license as well as appropriate accounts with the destination services.

10.2.1 Toolkit Compatibility

Templates that you create with the current Access Connector Toolkit are not backwards compatible with prior releases of the toolkit. You cannot import a connector from SocialAccess 2.3 into a toolkit that came with a prior version of SocialAccess. The import fails.

10.2.2 Provisioning Support

Provisioning is supported only through connectors created by NetIQ. At this time, you cannot create a custom connector template that supports provisioning user accounts to the connected system.

10.3 Federation Requirements for the Application Service Provider

As you explore the features of the Access Connector Toolkit, refer to the definitions in this section to understand the type of information you will need to collect from the destination web service or application.

assertion

A SAML 2.0 assertion is a package of identity attributes for an authenticated user that is sent from the trusted identity provider to the service provider.

assertion properties

The properties of the assertion include the following information:

- The recipient of the assertion.
- The LDAP identity attribute to use when federating users with the destination application service provider. Does the NameID require an email address format, or does it require unspecified format?
- The URL where SocialAccess should redirect the end user's session after the user logs in successfully with the URL provided on the connector configuration page.
- The binding method to use for identity information sent to the destination provider. For SAML 2.0, the only supported binding method is POST.

assertion attributes

The provider should provide a technical document that describes the attributes that are required for an assertion, such as the user's name or email address. It can include the attributes that are required to assign roles. The SAML assertion typically requires the nameID attribute. You must map the SAML assertion attributes to the matching attributes in your identity source.

entityID

The entityID is a field from the metadata that uniquely identifies that particular service provider, such as *sp_domain_name*.

For example:

google.com

The entity ID might use information from the federation instructions, or from a setting completed on the Configuration page when you deploy the connector.

federation instructions

The federation instructions provide the information that you will use to configure federation for SocialAccess on the service provider site. The information identifies where on the service provider's site to find the federation configuration capability as well as the field values and other guidance that you need to complete the required information.

When you configure the connector, the federation instructions will automatically provide the following information about your appliance as the identity provider:

- The URL for single sign-on

  `https://`*`appliance_dns_name`*`/osp/a/t1/auth/saml2/sso`

- The URL for single logout

  `https://`*`appliance_dns_name`*`/osp/a/t1/auth/app/logout`

- The URL for the identity provider's entityID

  `https://`*`appliance_dns_name`*`/osp/a/t1/auth/saml2/metadata`

- The X.509 signing certificate for the appliance

  The web service or application uses the certificate to set the trust relationship with SocialAccess.

  NOTE: When you copy the appliance's signing certificate, ensure that you include all leading and trailing hyphens in the certificate's Begin and End tags.

It provides the following information about your appliance if the login is initiated by the service-provider, such for connectors that use the WS-Federation protocol:

- The WS-Federation Passive URL
- The X.509 signing certificate for the appliance

metadata

The metadata is the configuration information that the application service provider uses to establish communications with the identity provider in an federation trust relationship. This usually includes a login URL or a customer-specific domain name, which is called the Assertion Consumer Service URL. Service providers allow you to export the required metadata to an XML file, or they provide the metadata in a public URL. The auto-generated metadata file from the service provider will not work as is. You must manually change the values to match your actual deployment environment.

The metadata usually includes the following information:

- The entityID for the service provider.
- The URL that receives the user identity information.
  - For SAML 2.0, the Assertion Consumer Service URL is where the assertion is posted by the browser. For example:

    `https://www.google.com/a/${`*`customer-domain`*`}/acs`

  - For WS-Federation, the Login URL is where the security token is posted by the browser. It corresponds to the `PassiveRequestorEndpoint` field from the metadata.
  - For SAML-In, the Single Sign-on Service URL is where the AuthnRequest will be posted. It corresponds to the `SingleSignOnService` field with a Post binding from the metadata.

    `https://accessmanager.base.url/nidp/saml2/sso`

- The logout URL corresponds to the `SingleLogoutService` field from the metadata.

- The logout URL Binding (HTTP Post or Redirect)

  The logout response URL
- The X.509 signing certificate

protocol binding

The protocol binding is the method used for transmitting assertions between the authenticating identity provider and the service provider. SocialAccess supports the Redirect and Post bindings for service-provider-initiated SSO, and the Post binding for identity-provider-initiated SSO.

nameID

The nameID is the attribute in the identity source that uniquely identifies the user. You must know whether this attribute requires the email address format or an unspecified format.

new settings

The new settings are appliance-specific settings that you want to allow the administrators to set when they configure the connector for an appliance.

For example:

- Customer-specific sections of the Assertion Consumer Service URL
- Connector-specific setting, such as a customer domain

security token

A WS-Federation security token is a package of identity attributes for an authenticated user that is sent from the trusted identity provider to the service provider. The provider should provide a technical document that describes the attributes that are required for the token, such as the user's name or email address. It can include the attributes that are required to assign roles.

signing certificate

The signing certificate is the X.509 certificate that identifies SocialAccess to the service provider. If you specify that the certificate is required by the service provider, the template automatically retrieves the appliance's certificate and inserts it in the Federation Instructions when you deploy the connector. You use the certificate when you set up the federated single sign-on for the application.

template properties

The template properties define the following information for the connector:

- Type of connector and type name (based on the template wizard)
- The unique name for the template file (target name)
- A brief description used as the connector name
- A 3-digit version number (ex: 1.0.0)
- A custom graphic to use for the icon that represents the connector in the SocialAccess administration console.

10.4 Creating a SAML 2.0 Connector Template

To create a connector for single sign-on with SAML 2.0, you can use the SAML2 option in the Access Connector Toolkit.

10.4.1 SAML 2.0 Requirements for the Application Service Provider

To create a custom SAML 2.0 connector, the application that connects to SocialAccess must meet the following protocol-specific requirements:

- ❒ Support identity federation using the SAML 2.0 protocol.

  For more information about SAML, see the OASIS website (https://wiki.oasis-open.org/security/FrontPage).
- ❒ Support the SAML web browser single sign-on profile, with the Redirect and POST bindings for service-provider-initiated SSO, and the POST binding for identity-provider-initiated SSO.
- ❒ Provide a capability in the application's administration console that allows you to enable and configure SAML SSO with SocialAccess as the identity provider.
- ❒ Provide technical documents that describe the application's SAML federation requirements, metadata, and assertions.

10.4.2 Planning for a SAML 2.0 Connector

Before you attempt to create the SAML 2.0 connector, you must collect information about the destination web service or application. For more information, see Section 10.3, "Federation Requirements for the Application Service Provider," on page 68.

Ask the application service provider the following types of questions to gather the required information:

- What does your SAML assertion look like?
- Do you have a SAML metadata document? What fields, if any, are customer-specific?
- Does your service support the SAML single logout protocol?
- What are the required configuration steps in your application to set up federation?
- What information do you provide to customers when they are setting up federation with their identity source?

10.4.3 Creating a SAML 2.0 Connector Template for an Application

A SAML 2.0 connector template consists of multiple components for federation, metadata, and assertion information.

To create a custom SAML 2.0 connector:

1. Log in as an administrator to the Access Connector Toolkit:

   `https://appliance_dns_name/css/toolkit`
2. Click **New** > **SAML2**.

   The connector **Type** is `SAML2`. The **Type Name** is `Generic SAML2 Connector`.
3. On the **Template** tab, complete the following information:
   - Template properties
   - Whether the service provider requires a signing certificate

- Federation instructions for the service provider
- New settings that need to be collected on the Configuration page of the connector

4 Click the **Metadata** tab, then use one of the following methods to specify the metadata:

- Select **Request**, then specify the source URL to retrieve the metadata.
- Complete the fields to manually generate the metadata.
- Import the values from a file or URL, and modify them for your deployment environment.

5 Click the **Assertion** tab, then define the properties and attributes required for the assertion.

5a On the **Properties** subtab, specify the properties for the assertion.

5b On the **Attributes** subtab, click **New**, specify and define the identity attribute, then click **Save**.

5c (Conditional) If the service provider requires other identity attributes for an assertion, repeat Step 5b to map the SAML assertion attribute to an attribute in your identity source.

6 (Optional) If it is supported, create the provisioning definitions. For more information, see Section 10.2.2, “Provisioning Support,” on page 68.

7 Click **Save** to save the new connector template.

8 Proceed to Section 10.8, “Exporting a Connector Template,” on page 76 to finish creating the new connector.

10.5 Creating a WS-Federation Connector Template

To create a connector for single sign-on with WS-Federation, you can use the WS-Fed option in the Access Connector Toolkit.

10.5.1 WS-Federation Requirements for the Application Service Provider

To create a custom WS-Federation connector, the destination application that connects to SocialAccess must meet the following protocol-specific requirements:

- ❒ Support identity federation using the WS-Federation protocol.

  For more information about WS-Federation, see the OASIS website (http://docs.oasis-open.org/wsfed/federation/v1.2/os/ws-federation-1.2-spec-os.html) or see the MSDN Library article (http://msdn.microsoft.com/en-us/library/bb498017.aspx).

- ❒ Support for the WS-Federation Passive Requestor Profile.
- ❒ Provide a capability in the application’s administration console that allows the customer to enable and configure WS-Federation SSO.
- ❒ Provide technical documents that describe the application’s WS-Federation federation requirements, metadata, and security tokens.

10.5.2 Planning for a WS-Federation Connector

Before you attempt to create a WS-Federation connector, you must collect information about the destination web service or application. For more information, see Section 10.3, “Federation Requirements for the Application Service Provider,” on page 68.

Ask the web service or application vendors the following types of questions to gather the required information:

- What does your WS-Federation security token look like?
- Do you have a WS-Federation metadata document? What fields, if any, are customer-specific?
- What are the required configuration steps in your application to set up federation?
- What is the information that you provide to customers when they are setting up federation with their identity source?

10.5.3 Creating a WS-Federation Connector Template for an Application

A WS-Federation connector template consists of multiple components for federation, metadata, and assertion information.

To create a custom connector:

1. Log in as an administrator to the Access Connector Toolkit:

   `https://appliance_dns_name/css/toolkit`

2. Click **New** > **WSFed**.

   The connector **Type** is `WS-Fed`. The **Type Name** is `Generic WS-Fed Connector`.

3. On the **Template** tab, complete the following information:
   - Template properties
   - Whether the service provider requires a signing certificate
   - Federation instructions for the service provider
   - New settings that need to be collected on the Configuration page of the connector
4. Click the **Metadata** tab, then use one of the following methods to specify the metadata:
   - Select **Request**, then specify the source URL to retrieve the metadata.
   - Complete the fields to manually generate the metadata.
   - Import the values from a file or URL, and modify them for your deployment environment.
5. Click the **Assertion** tab, then define the properties and attributes required for the security token.

   5a On the **Properties** subtab, specify the properties for the assertion.

   5b On the **Attributes** subtab, click **Predefined**, click the identity attribute, modify the definition if needed, then click **Save**.

   If a predefined option does not exist, use **New** to define it.

   5c (Conditional) If the service provider requires other identity attributes for an assertion, repeat Step 5b to map the WS-Federation attribute to an attribute in your identity source.
6. (Optional) Create the provisioning definitions. For more information, see Section 10.2.2, “Provisioning Support,” on page 68.

7 Click **Save** to save the new connector template.

8 Proceed to Section 10.8, “Exporting a Connector Template,” on page 76 to finish creating the new connector.

10.6 Creating a SAML 2.0 Inbound Connector Template

To create a connector for single sign-on with SAML 2.0 Inbound, you can use the SAML2 In option in the Access Connector Toolkit.

IMPORTANT: Connectors that you create in the Access Connector Toolkit using the SAML2 In option work only for users that you create on the SocialAccess appliance. SAML2 Inbound connectors will not work in SocialAccess with existing user accounts.

10.6.1 SAML2 In Requirements for the Application Service Provider

To create a custom SAML 2.0 Inbound connector, the destination application that connects to SocialAccess must meet the following protocol-specific requirements:

- ❒ Support identity federation using the SAML 2.0 protocol.

  For more information about SAML, see the OASIS website (https://wiki.oasis-open.org/security/FrontPage).

- ❒ Support the SAML web browser single sign-on profile, with the Redirect and POST bindings for service-provider-initiated SSO, and the POST binding for identity-provider-initiated SSO.
- ❒ Provide a capability in the application’s administration console that allows the customer to enable and configure SAML SSO.
- ❒ Provide technical documents that describe SAML federation requirements, metadata, and assertions.

10.6.2 Planning for a SAML2 In Connector

Before you attempt to create the connector, you must collect information about the destination web service or application. For more information, see Section 10.3, “Federation Requirements for the Application Service Provider,” on page 68.

Ask the web service or application vendors the following types of questions to gather the required information:

- What does your SAML assertion look like?
- Do you have a SAML metadata document? What fields, if any, are customer-specific?
- Does your service support the SAML single logout protocol?
- What are the required configuration steps in your application to set up federation?
- What is the information that you provide to customers when they are setting up federation?

10.6.3 Creating a SAML2 In Connector for an Application

A SAML2 In connector template consists of multiple components for federation, metadata, and assertion information.

To create a custom connector template:

1. Log in as an administrator to the Access Connector Toolkit:

   `https://appliance_dns_name/css/toolkit`

2. Click **New** > **SAML2 In**.

   The connector **Type** is `SAML2 In`. The **Type Name** is `Generic SAML2 In Connector`.

3. On the **Template** tab, complete the following information:
   - Template properties
   - Whether the service provider requires a signing certificate
   - Federation instructions for the service provider
   - New settings that need to be collected on the Configuration page of the connector

4. Click the **Metadata** tab, then use one of the following methods to specify the metadata:
   - Select **Request**, then specify the source URL to retrieve the metadata.
   - Complete the fields to manually generate the metadata.
   - Import the values from a file or URL, and modify them for your deployment environment.

5. Click the **Assertion** tab, then define the properties and attributes required for the security token.

   5a On the **Properties** subtab, specify the properties for the assertion.

   5b On the **Attributes** subtab, click **Predefined**, click the identity attribute, modify the definition if needed, then click **Save**.

   If a predefined option does not exist, use **New** to define it.

   5c (Conditional) If the service provider requires other identity attributes for an assertion, repeat Step 5b to map the WS-Federation attribute to an attribute in your identity source.

6. Click **Save** to save the new connector template.

7. Proceed to Section 10.8, “Exporting a Connector Template,” on page 76 to finish creating the new connector.

10.7 Modifying a Connector

You can modify the definition information for a connector by importing it in the Access Connector Toolkit. For example, you can import an existing connector to update its definition to the latest features available for connectors.

1. Obtain a copy of the connector’s ZIP file.

2. Log in as a SocialAccess administrator to the Access Connector Toolkit:

   `https://appliance_dns_name/css/toolkit`

3. Click **Import**, browse to and select the connector’s ZIP file, then click **OK**.

   The connector appears in the list of connector templates.

4. Click the **Edit** icon next to the **Display Name** for the connector template to open it in the Edit Connector Template window.

5 Modify the connector template settings as desired.

6 Click **Save** to apply the changes.

7 Click the **Export** icon next to the **Display Name** for the connector template.

8 Save the ZIP file for use on this or another SocialAccess system.

9 Proceed to Section 10.9, “Importing and Configuring Custom Connectors,” on page 76.

10.8 Exporting a Connector Template

After you create a connector template, you must use the Access Connector Toolkit to export it in a compressed ZIP file that you can import to any SocialAccess system. You then import the connector template in the SocialAccess administration console to make it available in the **Applications** palette.

To export the connector template:

1 Log in as a SocialAccess administrator to the Access Connector Toolkit:

`https://appliance_dns_name/css/toolkit`

2 Click the **Export** icon next to the **Display Name** for the connector template.

3 Save the ZIP file for use on this or another SocialAccess system.

4 Proceed to Section 10.9, “Importing and Configuring Custom Connectors,” on page 76.

10.9 Importing and Configuring Custom Connectors

SocialAccess allows you to import and configure custom connectors that you create with the Access Connector Toolkit, or that are created for you by NetIQ Technical Support or NetIQ partners.

After you export a custom connector, you must import its ZIP file to SocialAccess to make it available in the **Applications** palette of the administration console. Thereafter, you can enable and manage the connector as you do the connectors for applications that shipped with the appliance. The custom connector might require additional configuration, depending on the single sign-on method you use.

The destination application might also require additional configuration, depending on the application and the federation method.

To import and configure a custom connector:

1 Copy the custom connector ZIP file to the computer where you administer SocialAccess.

2 Log in as an administrator to the SocialAccess administration console:

`https://appliance_dns_name/appliance/index.html`

3 Click the **Tools** icon on the toolbar, then click **Import connector template**.

4 Browse to and select the custom connector ZIP file, then click **Import**.

5 Drag and drop the new custom connector from the **Applications** palette to the **Applications** panel.

6 (Conditional) For connectors that provide federated single sign-on, complete the connector settings on the **Configuration** tab.

The steps to configure the connector are determined by the information you added to the connector template.

7 (Conditional) For connectors that provide federated single sign-on, expand the **Federation Instructions**, then copy and paste the instructions into a text file to use when you configure the destination application.

NOTE: You must use a text editor that does not introduce hard returns or additional white space. For example, use Notepad instead of Wordpad.

---

8 Click the **Appmarks** tab, then review and edit the default settings for the appmark.

9 Click **OK** to save the configuration.

10 Click **Apply** to commit the changes to the appliance.

11 Wait until the configuration changes have been applied on each node of the SocialAccess cluster.

12 (Conditional) For connectors that provide federated single sign-on, configure the destination application for the appropriate federation method.

11 Customizing the End User Experience

After you have completed the configuration of the identity sources and the connectors for the SAML applications, you can perform additional customization such as rebranding, if appropriate.

11.1 Customizing Branding on User-Facing Pages

SocialAccess allows you to customize user-facing pages, such as the login page, so users see your company branding instead of the default NetIQ branding. After you have customized those pages, you can modify them as needed to meet new company requirements. Customizing the user pages does not affect any pages in the administration console itself.

If you implement custom branding in your SocialAccess environment and then re-run the initialization process to modify the DNS server or make other changes to an existing cluster, branding is reset to the default settings. Before you re-run the initialization process on an existing cluster, ensure that you back up your customized branding files so that you can reuse them.

IMPORTANT: Performing advanced branding customization requires advanced JavaServer Pages (JSP) knowledge. Before you make any changes, ensure that you have a good snapshot of your appliance that you can revert to if necessary. If you upload a bad branding file and are unable to log in to the administration console, you can re-run the appliance initialization to restore the default login pages. For more information, see Section 2.4, “Re-initializing the Appliance,” on page 17.

To customize branding for users:

1. (Conditional) If you plan to perform extensive rebranding, take a snapshot of the appliance.
2. Log in with an appliance administrator account to the SocialAccess administration console:

   `https://appliance_dns_name/appliance/index.html`
3. On the toolbar, click the Tools icon, then click **End user branding**.
4. (Conditional) If you want to customize the user login page, complete the following steps:
   - **4a** Click **Basic Customization**.
   - **4b** Change the title and background colors by specifying HTML color codes.
   - **4c** Change the default image by either not showing the image, or uploading a new image.

     The image size on the user landing and logout pages is always scaled to 60 x 60. On the login page, if the specified image is between a minimum of 60 x 60 and a maximum of 300 x 300, SocialAccess displays the image “as is.” If the image is outside this range, SocialAccess scales it to the nearest minimum or maximum value.
   - **4d** Click **OK** to save the changes and then click **Apply**.
5. (Conditional) If you want to perform more extensive rebranding, complete the following steps:
   - **5a** Click **Advanced Customization**.
   - **5b** Click **Download default end user login code**.
   - **5c** Save the file to your local computer.
   - **5d** Save a backup copy of the file.

5e Unzip the downloaded file and locate the `.jsp` files in the `osp\jsp` subdirectory.

5f Modify the desired JSP pages. The default text for the login page is located in the `osp\resources\oidp_custom_resources_en_US.properties` file.

For more information about the JSP files, see Appendix B, “Performing Advanced Branding,” on page 95.

5g Zip up the files again, but include only the `images` and `jsp` directories.

5h Log in to the administration console again.

5i On the toolbar, click the **Tools** icon, then click **End user branding**.

5j (Conditional) If you are customizing pages for the first time, click **Browse**, then browse to and select the modified file.

5k (Conditional) If you are updating previously customized pages, delete the name of the existing file. Click **Browse**, then browse to and select the `.zip` file that contains the newly modified `.jsp` files.

5l Wait until the file name changes to a hexadecimal value, then click **OK**.

5m Click **Apply**.

The pages now display the branding you customized in the `.jsp` files.

11.2 Configuring the Session Timeout for Users

After a user logs into SocialAccess, by default the session times out after 10 minutes of inactivity. You can change this session timeout value through the administration console.

To change the session timeout for users:

1 Log in to the administration console:

`https://`*`appliance_dns_name`*`/appliance/index.html`

2 Click the cluster icon in the lower left corner of the screen.

3 Click **Configure**.

4 Change the **User session timeout** field value. The value for the field is in minutes.

5 Click **OK**, then click **Apply** to save the changes.

12 Maintenance Tasks

SocialAccess allows you to change various appliance configuration settings as needed. For example, moving your appliance from a staging configuration to a production environment requires changes to the networking components.

12.1 Changing the Cluster Password

You can change the administrator password for the cluster as needed. The administrator password is the same for all nodes in the cluster.

To change the cluster password:

1. Log in with an appliance administrator account to the administration console:

   `https://appliance_dns_name/appliance/index.html`

2. Click the cluster icon at the bottom of the page, then click **Change cluster password**.
3. Type your old password, then type your new password twice and click **OK**.

12.2 Changing the IP Address

You can change whether a node uses DHCP or a static IP address.

1. Log in with an appliance administrator account to the administration console:

   `https://appliance_dns_name/appliance/index.html`

2. Click the node icon, then click **Configure**.
3. Select whether the appliance uses DHCP or a static IP address.

   If you select to use a static IP address, you can change the required values for the subnet mask, default gateway, and the DNS server.
4. Click **OK** to save the changes, then click **Apply** to apply the changes to the appliance.

If you have additional networking options configured, see Section 3.4, “Configuring Network Options,” on page 20.

12.3 Changing Public DNS Name or NTP Server Settings, or Uploading New Certificates

The appliance contains self-generated certificates. You can upload custom certificates through the administration console.

1. Log in with an appliance administrator account to the administration console:

   `https://appliance_dns_name/appliance/index.html`

2. Click the cluster icon, then click **Configure**.

3 Change the key pairs, NTP server, or public DNS name, then click **OK**.

4 Click **Apply** to apply the changes to the appliance.

Expired key pair certificates prohibit changes from being made to this page and make the key pair field red.

12.4 Updating the Appliance

SocialAccess provides an update channel for keeping your appliances up to date with the latest security fixes, bug fixes, and feature updates. Updates work only if you have registered each node in the cluster. For more information, see Section 3.3, “Registering SocialAccess,” on page 20.

When an update is available for one or more nodes in the cluster, the SocialAccess administration console displays a flag icon in the upper right corner of the window. You can also configure the appliance to send an email notification when an update is available. When you click the flag icon, you can see the version of the pending update, instructions on how to apply the update, and the Release Notes associated with the update patch.

The flag icon for the update channel appears only if you are logged in to the administration console with an administrator account. Other consoles do not display the flag icon.

SocialAccess automatically checks the NCC channel for updates once daily at 11:23:23 p.m. and downloads any available update. You can also manually check for updates any time by clicking **Tools > Check for updates** in the administration console. You can download and install an update as soon as the flag appears in the administration console, or you can wait for SocialAccess to download the update that night, to minimize network impact due to possible size of an update. NetIQ recommends always keeping your appliance up to date. However, updates are cumulative, so if you miss an update you can just install the next one when it is available.

IMPORTANT: If you apply an update to one node, you must apply the update to all the other nodes in the cluster. Update one node at a time. Ensure that the update was successful and the node is still working properly before you begin updating the next node. Do not perform any other administrative tasks requiring an **Apply** command, and do not switch the master node, until all nodes have been successfully updated to the same version of SocialAccess.

This process allows you to run in a mixed environment while updating each node. Once you have applied all available channel updates, the flag icon goes away.

To apply an update:

1 Take a snapshot of each node in the cluster to create a backup.

2 Click the appropriate node, then click **Apply update**.

SocialAccess displays status messages during the installation of the update and the rebooting of the node.

3 After the update completes and the node restarts, click **About** on the node to verify the updated version.

4 Verify the health of the updated node and all of the nodes in the cluster. Make sure all icons are green.

For more information, see Section 13.1, “Displaying Health,” on page 85.

5 Repeat Step 2 through Step 4 for each node in the cluster.

6 When you are sure all of the nodes in the cluster are working as expected, delete the snapshot.

12.5 Recovering from a Disaster

Use snapshots of the nodes to recover from a disaster. It is important to take snapshots of each node in the cluster regularly so you do not lose information.

To recover from a disaster:

1. On a regular basis, take snapshots of the nodes in the cluster.
   - **1a** Power off the working node, then take a snapshot. NetIQ recommends this method, but it requires that you shut down and restart the node in order to take the snapshot.

     or

     Take a snapshot of the running node, ensuring that you include the virtual machine's memory. Including the memory in the snapshot requires more time and space to store the snapshot, but taking a snapshot of a running node without the memory can result in corruption.
   - **1b** Repeat Step 1a for each node in the cluster, within a short time.
2. When a failure happens, restore the master node snapshot first.
3. Restore the other nodes in the cluster.

Use these steps only for disaster recovery. Never restore one snapshot. SocialAccess contains a database that is time-sensitive. Restoring only one node and not the others causes corruption in the appliance.

13 Troubleshooting SocialAccess

Use the information in the following sections to troubleshoot any issues you might encounter in SocialAccess.

13.1 Displaying Health

SocialAccess displays health for each node and for the cluster in the administration console. Hover the mouse over each node to display the health status of the node. If you want more details, click the node, then click **Show health.**

Show health displays the status for each component of the appliance. If the status is anything other than green (healthy), use the troubleshooting tools to determine what is wrong.

13.2 Troubleshooting Tools

SocialAccess provides troubleshooting tools if you encounter problems.

To access these tools:

1. Log in to the administration console:

   `https://appliance_dns_name/appliance/index.html`

2. Under **Appliances**, click the node, then click **Enter troubleshooting mode**.
3. Click the node again, then click **Troubleshooting tools**.
4. Select one or more of the troubleshooting scenarios listed.
5. Duplicate the error or condition.
6. Click **Download SocialAccess Log Files** to download the logs.

IMPORTANT: After you obtain the logs, turn off troubleshooting mode by clicking the node again and then clicking **Exit troubleshooting mode**. Leaving the logs running affects the performance of your appliance.

All of the log files in Table 13-1 are included in the download, no matter what scenario you select. The scenario that you select determines the amount of data that the log files display. Search the appropriate log file for errors while troubleshooting issues.

Table 13-1 Troubleshooting Log Files

Feature	Logs
Initialization or commands	ConfigurationReplicator.log
	ConfigurationReplicator_RL.log
	messages
	boot*
	packageoperations.log
	ag4c_configure.out
	ag4c.sh.out
Forward proxy	access.log
Administration console	adminui.log
Registration	register.log
Updates	zypper.log
	downloadUpdate.log
	afterUpdate.log
	beforeUpdate.log
	rpmsAfterUpdate.log
	rpmsBeforeUpdate.log
	rpmsUpdateDiff.log
	300_appliance_SnapshotUconPackages.sh.log
Custom Connectors	catalina.out
End User Authentication	catalina.out

13.3 Troubleshooting Different States

SocialAccess displays indicators for the current state of the different components. The display refreshes every five minutes. SocialAccess might not immediately display the change.

The following sections list the different components, the possible states, and troubleshooting steps to take when the state changes.

13.3.1 Front Panel of the Node

The indicator on the front panel of the node displays the health state of the node.

Figure 13-1 *Front Panel*

The states are:

Green: The node is healthy.

Yellow: The node cannot communicate with the other nodes within the five minute refresh.

Red: The node cannot communicate with the other nodes within two of the five minute refresh cycles.

Clear: The node is initializing or the state of the node is unknown.

Perform the following troubleshooting steps in the order listed if the state is anything but green:

1. Wait at least five minutes for the display to refresh and display the current state.
2. Click the node, then click **Show health**.

   Show health displays which part of the appliance is having issues.
3. If Show health displays a problem, use the troubleshooting tools to gather logs.

   For more information, see Section 13.2, “Troubleshooting Tools,” on page 85.
4. Restart the appliance, then wait at least another five minute cycle for all nodes to display the current state.

13.3.2 Top of the Node

The indicator on the top of the node shows whether the **Apply** commands completed successfully.

Figure 13-2 *Top of the Node*

The states are:

Green: All **Apply** commands completed successfully.

Red: The **Apply** commands did not complete successfully.

Perform the following troubleshooting steps in the order listed if the state is red:

1. Mouse over the top of the node to see the status of the last **Apply** command made on the node.
2. If there is not enough information in the summary, click **Enter troubleshooting mode** on the node, then mouse over the node again.

The troubleshooting mode displays a detailed summary for the last **Apply** command made on the node.

3. Reboot the appliance, then wait at least another five minute cycle for all nodes to display the current state.

13.3.3 Identity Source

The health indicator for the identity source is the small icon in the lower left corner.

Figure 13-3 *Identity Source Indicator*

The states are:

Green: The connector to the identity source is healthy.

Yellow: The connector has communication problems with the identity source.

Red: The connector to the identity source is unhealthy or contains errors.

Question mark: The state of the connector to the identity source is unknown.

Perform the following troubleshooting steps in the order listed:

1. If the connector is green, but the SocialAccess console is not displaying users, verify that the identity source servers are running and communicating properly.
2. Use the troubleshooting tools to gather logs, then look at the identity source provisioning logs listed in Table 13-1 on page 86 for errors. The `ConnectorLogs.txt` file maps the display name of the connector with the log name of the connector, if there is more than one identity source connector.
3. Click **Show health** on the master node, then expand **Operational**.

   If these items are yellow or red, the interface displays helpful information to help troubleshoot the issue.
4. If you are using LDAPS to communicate with the identity source, verify the LDAP certificates are not expired. You refresh the certificates as follows:
   a. Log in to the administration console, then click **Configure** on the identity source.
   b. Click the **Refresh** icon next to the identity source server.

13.3.4 Applications

The health indicator for an application connector is the small icon in the lower left corner.

Figure 13-4 *Application Indicator*

The states are as follows:

Green: The connector to the application is healthy.

Yellow: The connector to the application contains warnings.

Red: The connector to the application contains errors or cannot communicate with the application.

Question mark: The connector to the application is in an unknown state.

Perform the following troubleshooting steps in the order listed:

1. Click **Show health** on the master node, then expand **Operational**, and check the status of **Connectors**.
2. Use the troubleshooting tools to gather logs, then look at the logs listed in Table 13-1 on page 86 for errors.
3. Make a cosmetic change to the application connector configuration, then click **Apply**.

   By forcing an **Apply**, the appliance refreshes the application connector state and this can resolve the issue.

13.3.5 Tools

The health indicator for a tool is the small icon in the lower left corner. Only tools that report health have an indicator. The Google Analytics tool does not have a health indicator.

Figure 13-5 *Tool Indicator*

For all tools, the **Question Mark** icon indicates that the tool is in an unconfigured state.

Authentication Filter: The states for the Authentication Filter tool are as follows:

- **Green circle:** The connection to the destination ExtAPI script is healthy.
- **Red circle:** The connection to the destination ExtAPI script is not working. The ExtAPI script is unreachable.

Forward Proxy: The states for the Forward Proxy tool are as follows:

- **Yellow triangle:** The connection to or through the proxy is healthy. The triangle indicator serves as a warning that use of Forward Proxy is intended only for test environments.
- **Red circle:** The connection to or through the proxy is not working. The proxy device is unreachable.

Google reCAPTCHA: The states for the Google reCAPTCHA tool are as follows:

- **Green circle:** All of the configured identity sources are valid for use with reCAPTCHA.
- **Yellow triangle:** One or more of the configured identity sources are not valid for use with reCAPTCHA. For more information, see Section 3.6.1, “Requirements for reCAPTCHA,” on page 27.
- **Red circle:** None of the configured identity sources are valid for use with reCAPTCHA.

Syslog: The states for the Syslog tool are as follows:

- **Green circle:** The connection to the specified address:port is healthy.
- **Red circle:** The connection to the specified address:port is not working.

13.4 Troubleshooting Authentications

There can be multiple reasons why authentications to the SAML applications fail.

Time Synchronization: SocialAccess depends on timestamps to function correctly. Synchronize time between the VMware host, the appliance, and the workstations. Download the authentication logs. In the `catalina.out` file, search for the error `clock skew`.

SAML Authentications: Firefox contains a SAML debug add-on that you can use to view the SAML authentication between SocialAccess and the SAML applications. Download the add-on SAML tracer (https://addons.mozilla.org/en-US/firefox/addon/saml-tracer/) to view the SAML request.

Twitter Limitation: Twitter does not allow access to user email addresses, so SocialAccess cannot provide this information to the connector for NetIQ Access Manager. Users who log in to SocialAccess with a Twitter user account and then click an appmark for an Access Manager resource will receive the following error: `An Identity Provider response was received that failed to authenticate this session`.

A Custom Connector Worksheets

SocialAccess provides the NetIQ Access Connector Toolkit (ACT) that allows you to create custom connectors. If you need help creating a custom connector to use with SocialAccess, Priority Support customers have the option to open a service request with NetIQ Technical Support (NTS) (http://www.netiq.com/support). NTS is available to provide toolkit support as well as to configure the connectors to work with integrated applications. Additional information from the SaaS provider is usually required.

Before you contact NetIQ Technical Support, please complete the appropriate worksheet for the connector type that you want to create. The more information that you can provide, the better and more quickly NTS can help you create the connector.

A.1 Worksheet for SAML or WS-Federation Custom Connectors

For a SAML or WS-Federation custom connector, the destination service provider for the application is the trusted partner. Each connector requires information about how they support federation for the SAML protocol or WS-Federation protocol.

Table A-1 *Worksheet for a SAML or WS-Federation Custom Connector*

	Gather the following information:
❑	Which federation specifications will be used with various trusted partners? ❒ WS-Federation ❒ SAML 2.0 ❒ SAML 1.*x*
❑	Is the metadata (SAML/WS-Federation) from the trusted partner available?
❑	What profiles will you use to federate with your partners? ❒ WS-Federation Passive Requestor profile ❒ Browser POST profile ❒ Browser Artifact profile
❑	Is encryption of the assertions required? If so, which transport security protocols and certificates will be used?
❑	What user information is required by your partner for SSO? For example: email address, CN, and so on.

	Gather the following information:
❑	What name identifier format does your partner expect? ❒ Persistent ❒ Transient ❒ Email address ❒ Unspecified
❑	What attributes are required by your partner? Does a sample assertion exist from the trusted partner?
❑	To what URL on the partner side should an assertion or a claim be sent? (Assertion Consumer Service URL)
❑	To what URL on the partner side should a logout request be sent? (Logout URL and/or Logout Response URL)
❑	Do users need to be redirected to a specific application URL after an assertion has been successfully validated? (Destination URL)
❑	What are the contact details for the trusted partner (or partners), should we need to get them involved?
❑	All information needed by the trusted partner is available via the metadata at `https://`*`appliance_dns_name`*`/osp/a/t1/auth/saml2/metadata`

A.2 Worksheet for SAML In Custom Connectors

For a SAML Inbound (SAML In) custom connector, the identity provider is the trusted partner. Each connector requires information about how they support SAML federation.

Table A-2 *Worksheet for a SAML Inbound Custom Connector*

	Gather the following information:
❑	Which federation specifications will be used with various trusted partners? ❒ SAML 2.0 ❒ SAML 1.*x*
❑	Is the SAML metadata from the trusted partner available?
❑	What profiles will you use to federate with your partners? ❒ Browser POST profile ❒ Browser Artifact profile
❑	Which transport security protocols and certificates will be used? Assertions must be signed, and may be encrypted.
❑	What user information does the partner send for SSO? For example: email address, CN, and so on.

	Gather the following information:
❑	What name identifier format does your partner send with an assertion? ❒ Persistent ❒ Transient ❒ Email address ❒ Unspecified
❑	What attributes does your partner send? Does a sample assertion exist from the trusted partner?
❑	To what URL on the partner side should a logout request be sent? (Logout URL and/or Logout Response URL)
❑	What are the contact details for the trusted partner (or partners), should we need to get them involved?
❑	All information needed by the trusted partner is available via the metadata at `https://`*`appliance_dns_name`*`/osp/a/t1/auth/saml2/metadata`

B Performing Advanced Branding

SocialAccess allows you to customize user-facing pages, such as the login page, so users see your company branding instead of the default NetIQ branding.

IMPORTANT: Performing advanced branding customization requires advanced JavaServer Pages (JSP) knowledge. Before you make any changes, ensure that you have a good snapshot of your appliance that you can revert to if necessary. If you upload a bad branding file and are unable to log in to the administration console, you can re-run the appliance initialization to restore the default login pages. For more information, see Section 2.3, “Initializing the Appliance,” on page 17.

The brandable code is contained in a set of JSP files in the `/osp/jsp` directory.

To change the standard header or footer:

Edit the HTML formatting in one or both of the following files: `inc_common_body_top.jsp` or `inc_common_body_bottom.jsp`. The Java variables that the `inc_*.jsp` requires are listed at the top of the `inc_*.jsp` file. Much of this work can be done using the **Basic Customization** feature in the SocialAccess administration console. For more information, see Section 11.1, “Customizing Branding on User-Facing Pages,” on page 79.

To change the body (the space between the header and footer) of a page:

Each login, logout, and landing page contains the HTML formatting code that draws the “body” of its respective page. To customize this section of the page, you can edit the HTML formatting located between the include statements for the header and footer.

NOTE: The NetIQ logo at the bottom of the landing page cannot be removed or rebranded.

B.1 //Shared Include Files

inc_common_imports.jsp

Included in all non-shared JSPs in the Java imports area. Contains all shared import statements.

inc_common_java.jsp

Included in all non-shared JSPs in the Java code area. Contains the main Java code that processes request parameters and gathers data into standard Java variables that are used by the shared JSPs.

inc_common_head.jsp

Included in all non-shared JSPs in the JavaScript `<script>` area. Contains shared JavaScript functions that are used by the shared JSPs.

inc_common_body_top.jsp

Included in all non-shared JSPs at the top of the HTML `<body>` area. Contains the HTML formatting that creates the header for all non-shared pages. The header generally contains the product logo and name.

inc_common_body_bottom.jsp

Included in all non-shared JSPs at the bottom of the HTML `<body>` area. Contains the HTML formatting that creates the footer for all non-shared pages. The footer generally contains the NetIQ logo and a “demo version expired” warning, if applicable.

inc_common_locale.jsp

Included in the `loginselect.jsp` file in the Java code area. Contains Java code that sets the “temporary locale” form data items.

inc_common_usermessages.jsp

Included in all non-shared JSPs in the middle of the HTML `<body>` area. Contains the HTML formatting that creates the user error message section for all non-shared pages.

B.2 //Standard Login Page

loginselect.jsp

Contains the standard name-password form-based authentication page. All configurations of this page follow the standard header, body, and footer convention where the header and footer are defined in shared `inc_*.jsp` files. There are several optional objects on this page: radius login, recaptcha, and authentication contract selection.

B.3 //Second Factor Authentication Login Pages

logintotp.jsp

Contains the standard Timed One Time Password (TOTP) form-based authentication page.

fidoregister.jsp

Contains the standard Fast Identification Online (FIDO) form-based registration page.

fidologin.jsp

Contains the standard Fast Identification Online (FIDO) form-based authentication page.

B.4 //Landing Page (Home Page after Login)

landingpage_select.jsp

Contains the standard home page where users can select appmarks.

B.5 //Logout Page

landingpage_loggedout.jsp

Contains the “successful logout” page where the user is asked to close the browser to complete the logout.