SWIVEL®
AUTHENTICATION YOU CAN IDENTIFY WITH

PINsafe®

PINsafe
Manual
Version 3.1.3

Contents

Introduction

This document provides a general overview of PINsafe version 3.1.3, its key features and a quick start guide to the Administration console. It also covers what you should know about your installation of PINsafe and how to support and maintain your installation of PINsafe.

An important part of this document is the record of the installation, covered in Appendix C PINsafe Installation details. It is recommended that this section is completed at the time of installation and kept with the server or elsewhere where it can be found easily if required. A copy of the installation record needs to be sent to Swivel Secure for their records; this can be sent to

e-mail	**support@swivelsecure.com**
fax	**+44 1423 858172**

PINsafe 3.1.3 for PINsafe 3.1 users

This section describes the important new features of this version of PINsafe over the previous 3.1.x versions.

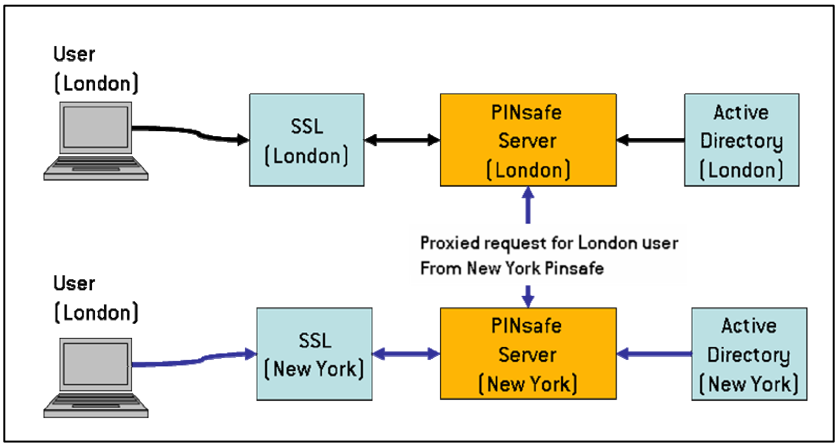
Peering and Proxying

It is now possible to deploy a number of PINsafe servers as a set of peers. Every user has an account on one of the PINsafe peers, however they can authenticate to any one of the PINsafe servers.

For example if a business has a London office and a New York office, running separate Active Directories and SSL VPNs a PINsafe server can be installed in each office. Each PINsafe server can be configured as a peer to the other.

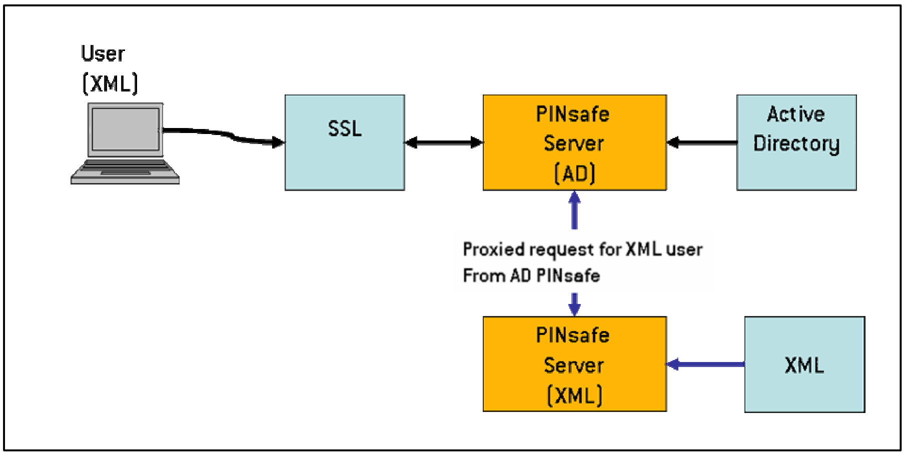
In this configuration, a London-based user can authenticate to the New York VPN; the New York PINsafe detects that the user is served by the London server and proxies the authentication request to the London PINsafe server.

The London PINsafe server checks the users credentials and returns the results to the New York server that, as a result, can allow or deny access.



Example of peering for multi-office installation

Another application of this is to support the inter-working of different user repositories. One PINsafe server can be configured to work with Active Directory, another to work with the XML repository. The two servers can be configured as peers so that a user from either user repository can authenticate to a VPN (or web application) authenticating via either PINsafe server.



Example of peering fro multi-repository installation

It is possible to deploy both these servers on the same PINsafe appliance, thus avoiding increased hardware costs.

Authentication Policies

A range of new policies are now supported by PINsafe that allow security managers to implement a range of rules.

These can be set at the server level and at the individual level. For example at the system level you can set a PIN expiry period and you can require a PIN to be changed at the first authentication after an admin has changed the PIN.

At the user level you can set PINs to never expire (that you may want for admin accounts) or to force a change PIN at first login.

This enables administrators to auto-provision users onto PINsafe, automatically generating and distributing PINs and then ensuring that users change their PINs at their first authentication.

Users can be informed of the need to change their PIN via an alert message, sent via SMS or e-mail; alternatively via a PINsafe authentication agent integrating with PINsafe via the Agent-XML API.

Logging

The logging capabilities of PINsafe have been significantly improved by the addition of SMTP and Syslog logging.

Syslog

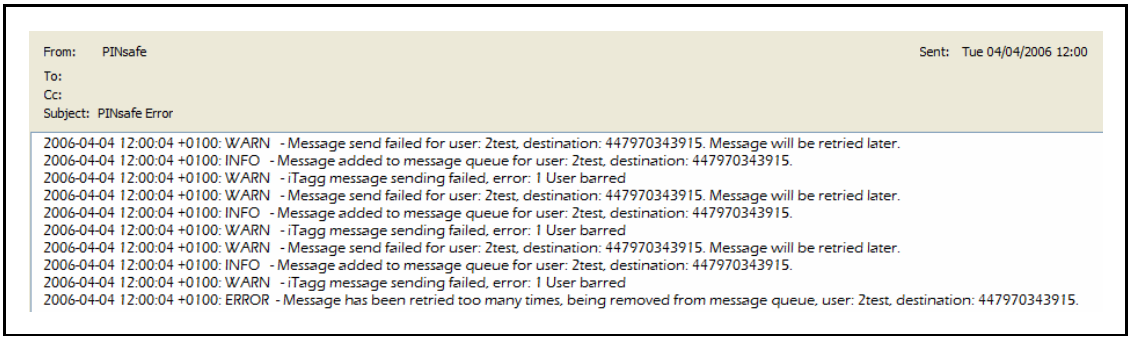
The support of logging via Syslog means that log files can be written to a remote server using the *de-facto* standard for system event logging, Syslog. Syslog is supported as standard in Unix/Linux environments and can be supported by free third party Syslog packages in Windows ™ environments.

Using syslog provides a number of advantages:-

- PINsafe logs can be handled in a consistent way to other customer systems.
- Remote logging means that logs are maintained if there is a hardware failure on the PINsafe appliance.
- Logs cannot be overwritten my any malicious attack on the PINsafe server.

SMTP

STMP logging means that PINsafe can be configured to email certain events, eg server errors or account lock-outs, to designated email addresses. This gives the managers of the system warning of potential server issues, to allow them to be addressed in a timely manner, reducing impact on the end-users.



From: PINsafe　　Sent: Tue 04/04/2006 12:00
To:
Cc:
Subject: PINsafe Error

```
2006-04-04 12:00:04 +0100: WARN  - Message send failed for user: 2test, destination: 447970343915. Message will be retried later.
2006-04-04 12:00:04 +0100: INFO  - Message added to message queue for user: 2test, destination: 447970343915.
2006-04-04 12:00:04 +0100: WARN  - iTagg message sending failed, error: 1 User barred
2006-04-04 12:00:04 +0100: WARN  - Message send failed for user: 2test, destination: 447970343915. Message will be retried later.
2006-04-04 12:00:04 +0100: INFO  - Message added to message queue for user: 2test, destination: 447970343915.
2006-04-04 12:00:04 +0100: WARN  - iTagg message sending failed, error: 1 User barred
2006-04-04 12:00:04 +0100: WARN  - Message send failed for user: 2test, destination: 447970343915. Message will be retried later.
2006-04-04 12:00:04 +0100: INFO  - Message added to message queue for user: 2test, destination: 447970343915.
2006-04-04 12:00:04 +0100: WARN  - iTagg message sending failed, error: 1 User barred
2006-04-04 12:00:04 +0100: ERROR - Message has been retried too many times, being removed from message queue, user: 2test, destination: 447970343915.
```

Example Error e-mail log

The email contains the ERROR event in the logs along with the preceding log entries to aid diagnosis.

Improved User Administration

Improvements have been made to the way users are administered within PINsafe. Accounts can be set to disabled; so that they users can be prevented from authenticating but their details remain on the server.

The user-interface on the admin console now allows for searching by account status; making it easier to identify locked/disabled user accounts; reducing administration overheads.

The navigation has been made cleaner and easier to navigate.

Upgrade

Release 3.1.3 is a full release. Uses of previous versions can upgrade to this version to access the new capabilities; there are no critical bug-fixes in this release therefore upgrade is not mandatory

3.1.3 extends the agent-XML API; to make best use of the new features customers are advised to update their agents and filters to new versions that exploit these features.

Upgrade instructions and new filters/samples will appear on the Swivel Partner Resource Website.

Upgrading from 3.1

In the following instructions, x.x refers to the Tomcat version number and a safe location means one not under the Webapps directory that is likely to get overwritten when the new .war file is deployed.

To upgrade from Versions 3.1, 3.1.1.

1. Stop the Tomcat Server
2. It is **highly recommended** that you take a copy of the entire ...\Tomcat x.x\webapps\pinsafe\WEB-INF\ folder but specifically ensure steps 3,4,5 and 6 are completed
3. Back up the PINsafe configuration by taking a copy of

...\Tomcat x.x\webapps\pinsafe\WEB-INF\conf\config.xml to a safe location

4. Back up the PINsafe User Repository by taking a copy of

...\Tomcat x.x\webapps\pinsafe\WEB-INF\data\repository.xml to a safe location

5. Back up the PINsafe user data by taking a copy everything under

...\Tomcat x.x\webapps\pinsafe\WEB-INF\db to a safe location

6. Back up any customized transport, user repository classes residing on the PINsafe server.
7. Delete the pinsafe directory from the webapps folder
8. Copy the .war file to the webapps folder
9. Start the Tomcat server, this will deploy the PINsafe application.
10. Stop the server
11. Restore the files you saved in steps 3,4,5,6 back to their original locations.
12. Restart the server

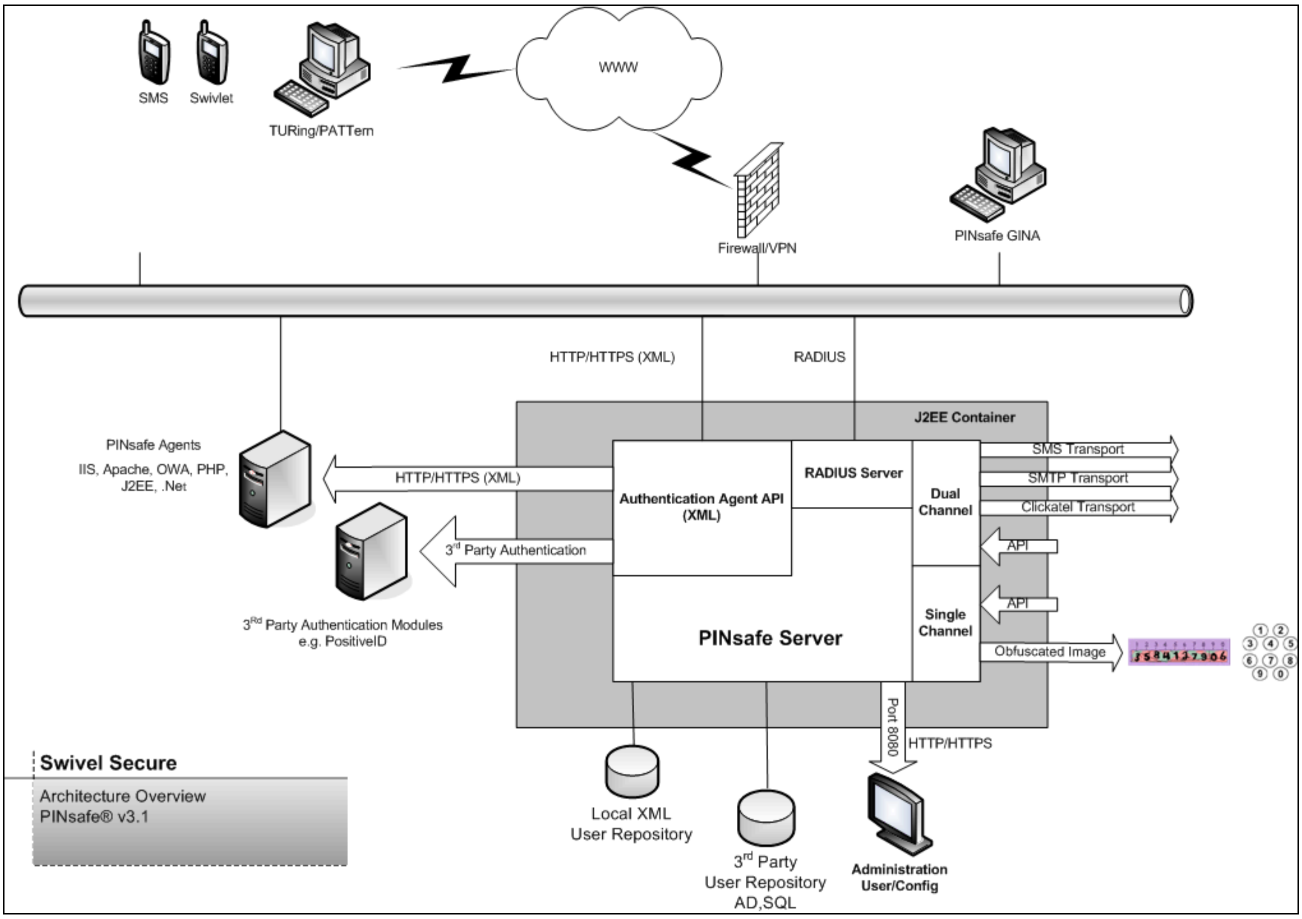
For upgrading form pre 3.1 versions consult support@swivelsecure.com

Architecture

The PINsafe application runs within a JSP/Servlet Container, Apache Tomcat being the default that is recommended. It also requires a Java JDK. In addition for certain configurations of the product the Java Communication API needs to be installed; more details in the installation section.

PINsafe can use its own database for storing user accounts or can be configured to work with an existing Active Directory or other external user repository; see “How to guides” in the Appendix.

However there is an additional component required if using the GSM transport and that is the Java Communications API. Links to these components can be found in the installation section.



PINsafe Architecture

PINsafe Installation

An important part of this document is the record of the installation, covered in Appendix C PINsafe Installation details. It is recommended that this section is completed at the time of installation and kept with the server or elsewhere where it can be found easily if required. A copy of the installation record needs to be sent to Swivel Secure for their records; this can be sent to

e-mail	**support@swivelsecure.com**
fax	**+44 1423 858172**

PINsafe can be purchased as an Appliance or software only. If an Appliance is shipped with PINsafe, all software will come pre-loaded. Please note that the installation should only be attempted by someone comfortable with this kind of installation. Also if you install PINsafe on your own hardware platform and OS that you should take all the necessary steps to security harden the installation.

Before you deploy and start PINsafe, ensure you have the following software installed:

- Java runtime environment (JRE) Version 5, available from www.sun.java.com
- Tomcat v5 available from http://tomcat.apache.org/
- Java Communications API if you intend using a GSM MODEM (follow the Java Comm API installation instructions) See Appendix A for instructions. This is available from http://java.sun.com/products/javacomm/downloads/index.html
- The PINSafe distribution (.war) file

Server Configuration

Once all the software has been installed on the server it has to be configured to work within the environment in which is to be installed. The following instructions are for a RED HAT LINUX installation; for a Windows installation please contact support@swivelsecure.com

Network Configuration

The default IP address for the machine is 192.168.10.10. The default hostname is changeme.swivelsecure.com.

The easiest way to do changes this is via X Windows. (To launch X-Windows type startx in a terminal window) Click on Applications -> System Settings -> Network.

Select the Active Ethernet interface and click on Edit. Change the IP, subnet, and gateway to reflect your network. Click on the "DNS" tab to change the DNS entries, including search path, and click on the "Hosts" tab then Edit to change the hostname and the first alias listed. Click on File -> Save to save these changes.

After saving the changes you will either need to reboot the machine, or, at a prompt, type:

```
service network restart
```

SSL Configuration

SSL configuration should be done on-site in order to ensure that the SSL certificates are set up properly for the client.

A) Create a local Certificate Signing Request (CSR)

In order to obtain a Certificate from the Certificate Authority of your choice you have to create a so called Certificate Signing Request (CSR). That CSR will be used by the Certificate Authority to create a Certificate that will identify your website as "secure". To create a CSR follow these steps:

Create a local Certificate:

```
keytool -genkey -alias tomcat -keyalg RSA -keystore <your_keystore_filename>
```

Note: In some cases you will have to enter the domain of your website (i.e. www.myside.org) in the field "first- and lastname" in order to create a working Certificate.

The CSR is then created with:

```
keytool -certreq -keyalg RSA -alias tomcat -file certreq.csr -keystore
<your_keystore_filename>
```

Now you have a file called certreq.csr that you can submit to the Certificate Authority (look at the documentation of the Certificate Authority website on how to do this). In return you get a Certificate.

B) Importing the Certificate

Now that you have your Certificate you can import it into you local keystore. First of all you have to import a so called Chain Certificate or Root Certificate into your keystore. After that you can proceed with importing your Certificate.

Download a Chain Certificate from the Certificate Authority you obtained the Certificate from.
For Verisign.com go to: http://www.verisign.com/support/install/intermediate.html
For Trustcenter.de go to: http://www.trustcenter.de/certservices/cacerts/en/en.htm#server
For Thawte.com go to: http://www.thawte.com/certs/trustmap.html

Import the Chain Certificate into you keystore

```
keytool -import -alias root -keystore <your_keystore_filename> -trustcacerts -file
<filename_of_the_chain_certificate>
```

And finally import your new Certificate

```
keytool -import -alias tomcat -keystore <your_keystore_filename> -trustcacerts -
file <your_certificate_filename>
```

You will also need to tell Tomcat to respond to SSL requests. In order to do this, go into the Tomcat server.xml file with your editor:

/usr/local/<Jakarta-tomcat-directory>/conf/server.xml

Uncomment the section for your SSL configuration. If it is using SSL on a non-standard port (for example, 8443) make sure to make a note of this. You will then need to restart Tomcat in order to make this configuration go live.

IPtables Editing

By default, the iptables rules are set up to allow incoming connections on only 3 ports:

1. TCP port 22 (SSH)
2. TCP port 8080 (HTML / HTTP)
3. TCP port 443 (SSL-HTML / HTTPS)

If you have configured Tomcat to use a different port, such as 80 or 8443, you will need to edit the iptables rules. The easiest way to do this is via the command line. As root, edit the file /etc/sysconfig/iptables.

The file will look something like this:

```
# Firewall configuration written by redhat-config-securitylevel

# Manual customization of this file is not recommended.

*filter

:INPUT ACCEPT [0:0]

:FORWARD ACCEPT [0:0]

:OUTPUT ACCEPT [0:0]
```

```
:RH-Firewall-1-INPUT - [0:0]
-A INPUT -j RH-Firewall-1-INPUT
-A FORWARD -j RH-Firewall-1-INPUT
-A RH-Firewall-1-INPUT -i lo -j ACCEPT
-A RH-Firewall-1-INPUT -p icmp --icmp-type any -j ACCEPT
-A RH-Firewall-1-INPUT -p 50 -j ACCEPT
-A RH-Firewall-1-INPUT -p 51 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 443 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 8080 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 1812 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 1813 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
```

The important lines are at the end, just before the second to last line. In your editor, copy the last "ACCEPT" line and edit the "--dport" section to include the port number you need open. Save and close the file.

At the prompt, run:

```
service iptables restart
```

This will restart the iptables service and cause your changes to go into effect.

NOTE: Always try to edit iptables locally. It is very easy to make a mistake in configuration and lock yourself out on a remote connection.

Remote User Accessibility

By default, the only way to remotely administer the machine is via an SSH connection to port 22. This will give you a console window.

The root user is not allowed to SSH in remotely. An administrator should log in as the "swivel" user, then use the "su" command to become root.

The SSH configuration file is located at /etc/ssh/sshd_config . If for security reasons it is deemed necessary to change what port SSH is listening on, it can be changed in there. For example, many sites have SSH listening on port 2222 instead of 22 so that the daemon can be run as a non-trusted user. This would be changed by editing the following line:

```
#Port 22
```

to read:

```
Port 2222
```

Then, after saving the file, doing a "service sshd restart". Note that you will also have to update the iptables rules (see above) in order to open up the proper port.

If there are multiple administrators, or you wish to limit some administrators to specific functions, the "sudo" command is installed. Simply run "visudo" as root and assign the appropriate users the privileges they need. They will need to run "sudo" before those commands and enter in their own password. Some examples (from the sudoers file):

```
# User privilege specification
root    ALL=(ALL) ALL

# Uncomment to allow people in group wheel to run all commands
# %wheel      ALL=(ALL)     ALL

# Same thing without a password
# %wheel      ALL=(ALL)     NOPASSWD: ALL

# Samples
# %users  ALL=/sbin/mount /cdrom,/sbin/umount /cdrom
# %users  localhost=/sbin/shutdown -h now
```

Registration of the OS

If you purchased PINsafe with an appliance it comes with a year's subscription to the Red Hat Network. This give access to the Red Hat up2date feature that enables OS updates to be easily installed on the machine.

Setting up up2date

The Red Hat Network function "up2date" will automatically update the machine with Red Hat provided packages.

From a prompt, type "up2date" or double-click on the red pulsing "!" icon on the desktop.

The up2date program will then walk the users through the set up procedure and begin the update of the machine.

If the user wishes to automate the procedure, they can do so by setting up a cronjob to handle it. As root, type crontab –e. In the editor, add the following line:

```
5 0 * * * up2date-nox –uv
```

This will run up2date at 5 minutes past midnight every day. Since Red Hat updates their packages frequently, this is a good default configuration. If this is too often, check the crontab (5) documentation for how to set it for a specific configuration, or run it manually via the GUI interface.

However,it is not recommended that this is run as a cron job as there is risk (albeit very small) that a Red Hat update will affect the operation of PINsafe. Swivel will constantly test Red Hat updates to ensure that PINsafe is not affected; You should contact support@swivelsecure.com for advice on specific Red Hat updates.

PINSafe Deployment

To deploy and start PINsafe perform the following steps:

1. Place the war file in Tomcat's webapps folder
2. Start Tomcat
3. Browse to http://PINsafe_server_IP_address:8080/pinsafe
4. The default administrator is 'admin' with a PIN of '1234'.

Creating an Admin User

The default administrator account is only available until a user repository is selected; so it is important to create an admin user account once start working with a user repository

How to create an Admin account with the XML Repository

Go to Repository > General

From the drop down options select XML, then click Apply

Go to User Administration (Left Hand Column)

Click Add User

Enter the user details; remember to make a note of the username that you have entered.

Select the PINsafeAdministrators and PINsafeUsers check boxes

Click apply

Go back to the User Administration screen

Select Sync Now

Select RESET PIN for the account and enter a new PIN. Make a note of this PIN

It is recommended that you open a new browser window, navigate to the admin console and log on using this new account before you exit the existing admin console session

How to create an Admin account with the Active Directory Repository

Go to Repository > General

From the drop down options select Active Directory

Go to Repository > Groups

Against Administrators, enter the fully distinguished name of the Active Directory group to which PINsafe administrators will belong and click apply. (See Appendix B)

Against PINsafe User, enter the fully distinguished name of the Active Directory group to which PINsafe users will belong and click apply.

Administrators of PINsafe must be members of both these groups; ensure that there is at least one account that meets these criteria

Go to Repository > Active Directory

Configure PINsafe to synchronize with your Active Directory server.

Go to User Administration and select Sync Now

This will then create PINsafe accounts associated with the active directory accounts, including PINsafe accounts with admin rights for active directory accounts that belong to the relevant group.

Search to find the administrator account that you have created.

Select RESET PIN for the account and enter a new PIN. Make a note of this PIN

It is recommended that you open a new browser window, navigate to the admin console and log on using this new account before you exit the existing admin console session.

License Key

PINsafe comes with a 5 user evaluation license. To operate a live PINsafe server you need a valid licence key obtained from your reseller or from Swivel Secure. Once you have this license key enter this key on the server > license screen.

Integration

The integration tasks for PINsafe depend on the specifics on the installation and the technology with which PINsafe is required to operate. There is a number of potential integration points including;-

Agents

Agents are the integration points between PINSafe and what it is that PINSafe is being used to protect. There are agents that can be used to integrate PINsafe with Web-servers with VPNs.

Radius

PINsafe also can act as a RADIUS server, so an alternative method of integration is to configure PINsafe to accept authentication requests from a RADIUS (NAS) client.

User Repository

PINsafe come with its own user repository but it can also be configured to work with existing external user repositories such as Active Directory

Third Party Authentication

PINsafe is an open authentication platform in that it can be used in conjunction with other authentication technologies. PINsafe has a third party API to support this interaction

Transport

PINsafe sends security strings to end user via a transport layer implemented by a transport class. Different classes allow for security strings to be sent via different methods, eg SMTP, SMS. Transport classes can also be configured to send alert information to users, eg to inform them that their PINsafe account has been created.

These integration points are described in more details below.

Integrating with Agents

An agent is a piece of software residing outside the PINsafe platform but which communicates with the PINsafe platform to manage authentication. There are a number of ready made agents available and new agents can generally be created very easily. This section covers how PINsafe can be integrated with agents to support using PINsafe with a web application and with a VPN.

How to use the custom attribute

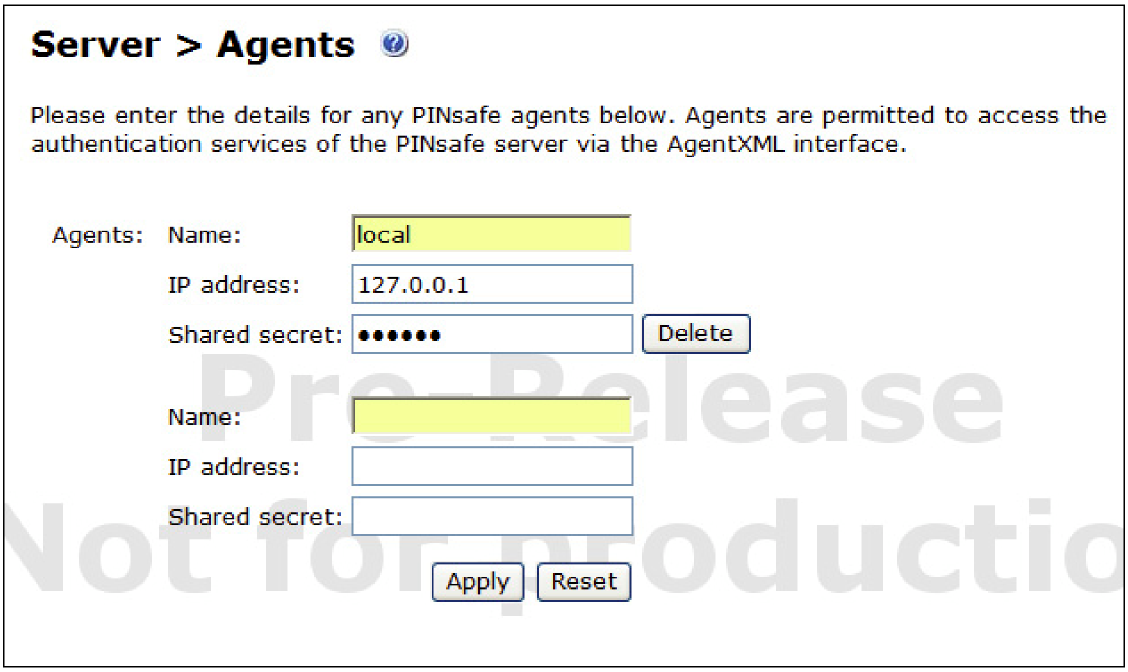
It is possible to assign a custom attribute to a user from the user admin screen. When a user successfully authenticates via the Agent-XML interface, this custom attribute is returned. This can be used to add granularity to the access rights granted on successful authentication.

How to integrate PINsafe with an external agent.

The PINsafe server will only service authentication requests from trusted agents.

The configuration of PINSafe to work with an external agent therefore consists of adding the details of the agent to the PINsafe trusted agent list. To do this go to the Agents screen, by clicking on Agents in the left hand navigation bar. Then enter a name for the agent, the IP address of the agent and then a shared secret.

In order for authentication request to be processed the source IP address of the request and the shared secret presented by the agent need to match the details entered on this screen.



Server > Agents

Please enter the details for any PINsafe agents below. Agents are permitted to access the authentication services of the PINsafe server via the AgentXML interface.

Agents: Name: local

IP address: 127.0.0.1

Shared secret: ●●●●●● Delete

Name:

IP address:

Shared secret:

Apply Reset

Screen shot of Agent Configuration Screen

Then click apply

How to integrate PINSafe with an IIS Website.

In this instance the Agent is implemented using an ISAPI Filter. This filter needs to be installed on the Webserver hosting the site to be protected. For information about filtering within IIS go to the Microsoft web site. http://msdn.microsoft.com/library/default.asp?url=/library/en-us/iissdk/html/22e3fbfb-1c31-41d7-9dc4-efa83f813521.asp

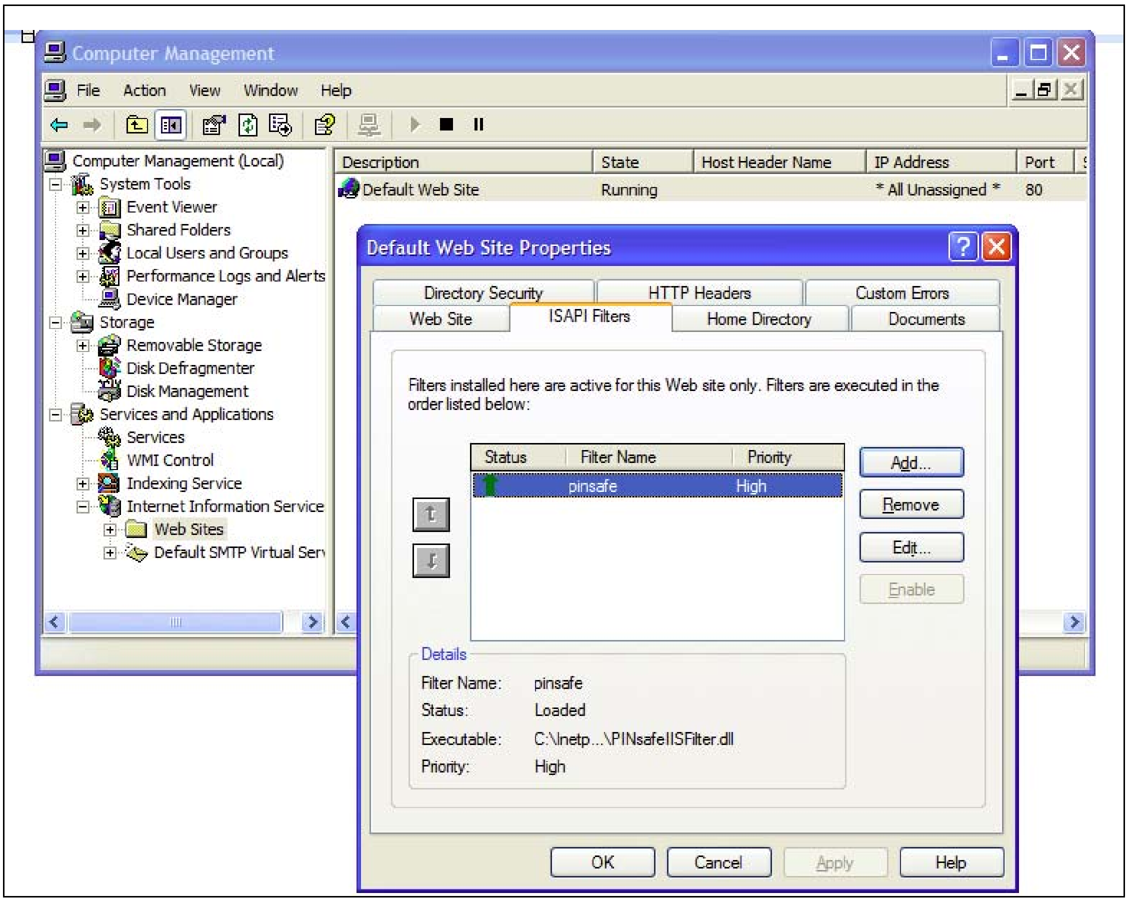
To install the Pinsafe ISAPI filter you need to run the install file PINsafeIISFilter.msi on the IIS server. This will install all the required files onto the server under the Inetpub\wwwroot\PINsafe directory.

The configuration of the filter is achieved by editing the web.config file which has the following name-value pairs.

Name	Value
PINsafe_SSL	True if SSL is being used between the agent and PINsafe, otherwise false
PINsafe_Server	The IP address or hostname of the PINsafe server, which must be visible to IIS server
PINsafe_Port	The port that the PINsafe server services requests on; usually 8080
PINsafe_Context	The web application context in which the PINsafe server is installed.
PINsafe_Secret	The shared secret between this agent and the PINsafe server. Needs to match value entered as part of PINsafe config
PINsafe_IdleTime	The connection idle time (in seconds) after which re-authentication will be required.
PINsafe_CookieSecret	The secret used to hash the authentication cookie. This should be a long, random string that is very difficult to guess.
PINsafe_ExemptAddress	Comma separated list of IP addresses that are exempt from the requirement to perform PINsafe authentication. Partial addresses such as "192.168" may be entered.
PINsafe_ExemptPath	Comma separated list of paths that are exempt from the requirement to perform PINsafe authentication. Partial paths may be entered, for example "/images/" would allow access to "/images/logo.png" and "/images/staff.jpg"
PINsafe_EnforcedPath	Comma separted list of paths that require PINsafe authentication. If any paths are entered, then only those paths will be subject to PINsafe authentication and any path exemptions will be ignored. Partial paths may be entered.
PINsafe_LogoutPath	A path that causes the cookie to be deleted and authentication to be required on any subsequent request.
PINsafe_LoginPath	The path that users will be redirected to when they are required to perform PINsafe authentication. Default is /PINsafe/login.asp but user can produce their own customised version of this page.
PINsafe_LoginStylePath	The path that the login form loads it's style information from.
PINsafe_ImagePath	The path from which the login form loads single, channels images. Default being /PINsafe/image.asp

Table of IIS configuration parameters

To apply the filter you need to configure the web site properties.



Description	State	Host Header Name	IP Address	Port
Default Web Site	Running		* All Unassigned *	80

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

Status	Filter Name	Priority
	pinsafe	High

Details

Filter Name: pinsafe
Status: Loaded
Executable: C:\Inetp...\PINsafeIISFilter.dll
Priority: High

Screenshot of IIS filter installation

Once the filter has been applied to the website, when a user attempts to access part of the website protected by PINsafe, PINsafe will prompt them to authenticate

How to write an Agent

A more flexible approach than using a web filter is to develop an agent specifically for a web site or web application. PINsafe is an open platform that allows integration to other systems via a set of APIs. Swivel provide a range of samples to help support this integration, for example how to make authentication requests from within a jsp. For more details e-mail support@swivelsecure.com

Integrating with RADIUS

PINsafe can operate as a RADIUS server for working with external systems such as SSL VPNs.

How to configure PINsafe to operate as a RADIUS server.

In order to integrate PINsafe with an external system the PINsafe RADIUS server first needs to be enabled and then configured to receive authentication requests from the external systems, or Network Access Server (NAS) using RADIUS terminology.

The RADIUS server is configured by going to the Radius-> Server config page. Enter the required configuration details and then select Apply.

Once the Radius Server is configured and enabled, it can be configured to accept requests from a NAS. This is achieved by selecting the Radius->NAS page and entering the details.

For authentication requests to be processed by PINsafe they need to come from the IP address specified in the NAS configuration, the shared secret needs to match that specified in the NAS configuration and be sent to the IP address specified in the PINSafe Server Configuration.

How to integrate a VPN with PINsafe using RADIUS.

In order to configure a VPN to use PINsafe for authentication the VPN needs to be configured to make authentication requests against the PINsafe server. This method for doing this depends on the VPN concerned so you need to consult the relevant documentation for the specific VPN. Swivel Secure will be able to provide you with specific instructions for a range of VPNs.

The basis of the configuration is basically to ensure that.

1. The VPN server needs to send authentication requests to the IP address and port number on which the PINsafe Radius server is operating.
2. The VPN server needs to have its shared secret (and realm, if specified) set to match that on the NAS configuration screen.

This is all the configuration that is required for use with a VPN, however , if TURing is being used and you require the TURing image to be integrated into the VPN log on page; the log-on page needs modification.

How to use single channel with a VPN

In order to use single channel authentication with a VPN, the user needs to be able to obtain a TURing image, perform the OTC extraction using their PIN and then enter these details onto he VPN authentication page.

One way of delivering a security string to the user is to modify the VPN log-on page so that it incorporates a TURing image; like the example below. Some VPNs accommodate this form of integration; others require some customization and others make this impossible. PINsafe have a number of these customizations available off the shelf.

However there are a number of ways of delivering the security string in a TURing image that do not require the VPN log-on page to be modified and thus are a very low-touch integration.

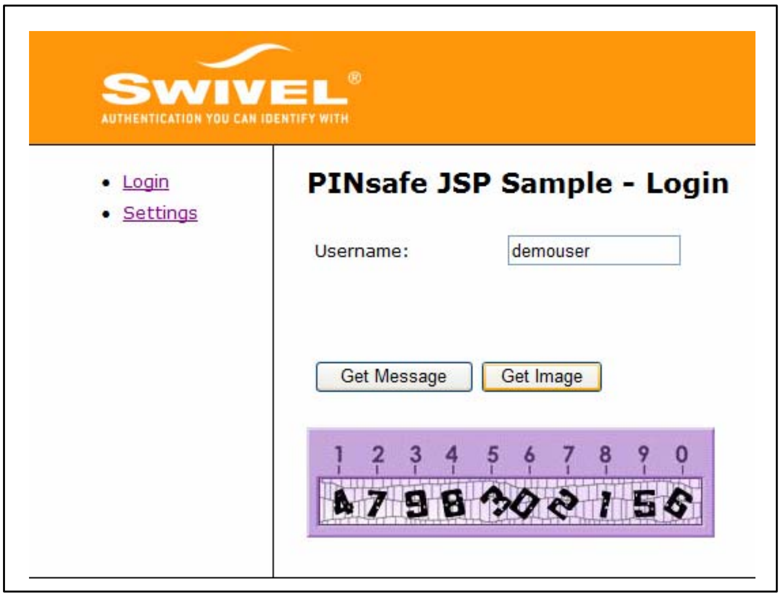
User Portal

One solution to delivering the security strings is to use a user-portal web application to allow the user to obtain a TURing image via a web browser.

Swivel can supply a user-portal is can be hosted on any servlet container, including the server on which PINsafe is running. It acts as an authentication agent for PINsafe and can be used to deliver security strings to the end-user and also for functions such as allowing the user to change their PIN numbers. It is easily branded and customised.

The beauty of this solution is that it does not require and client to be installed on the user's laptop, all they need is a standard browser.

Users can bookmark this page and even have it on their desktop as a bookmark. Users can be sent the url of the portal as part of the provisioning of PINsafe.



PINsafe JSP Sample - Login

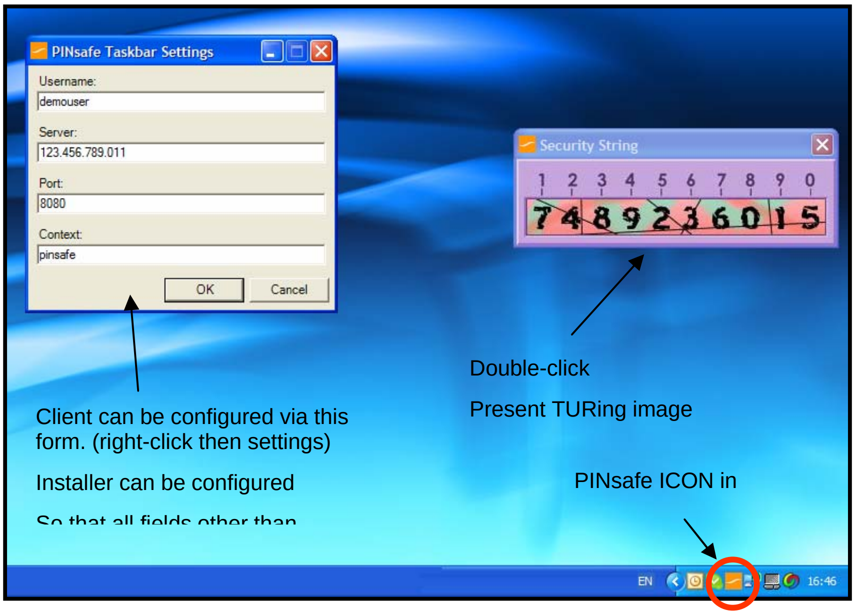
Username: demouser

User Portal

To authenticate the user opens a browser at the portal's url, enter their username (although this could be pre-filled, stored in a cookie). They select get image and the image is presented to them. The user performs the one-time code extraction and enters the one-time code into the VPN log-on form.

SysTray Utility

Another way to deliver the TURing image to the end-user is to use the PINsafe SysTray utility. This is a small client that resides in the users System Tray.



Username: demouser

Server: 123.456.789.011

Port: 8080

Context: pinsafe

Client can be configured via this form. (right-click then settings)

Installer can be configured

So that all fields other than

Double-click

Present TURing image

PINsafe ICON in

Screenshot of Systray Utility

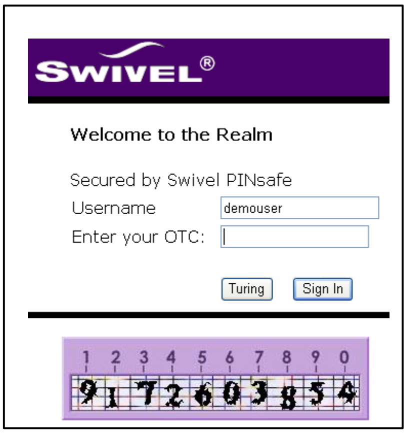
The application can be manually configured to retrieve TURing images from any PINsafe server, for any user. It can also be pre-configured so that the user need only enter their username. Once the application is configured, users can obtain a security string merely by double-clicking on the PINsafe icon in the system tray (or by right-clicking and selecting get image)

The application is small and easy to distribute; it can be provided with its own windows installer.

The user of the sys-tray application gives a true soft-token like experience.

These two options are not mutually exclusive a single PINsafe server can accommodate both alternatives.

How to incorporate a TURing image into a VPN log on page.



Welcome to the Realm

Secured by Swivel PINsafe

Username demouser

Enter your OTC:

If you wish to integrate a TURing image with a VPN log on page then the VPN configuration needs modification.

The implementation of these modifications again depend on the VPN itself, some VPNs allow you to easily customize the log on page, for others it is more invovled.

The log on page needs to

1. Have a "Start Session" or "TURing" button.
2. The user enters their username and then selects this button
3. This then fetches the Turing Image; this will be in the format
4. http://PINsafe IP Address:8080/pinsafe/SCImage?username="demouser" It is recommended that the request for the image is proxied via the VPN or via other means so that port 8080 on the PINsafe server does not need to be opened up to the internet.
5. The user can then enter their OTC and select Sign In; this will then allow the VPN to use the username and extracted OTC to authenticate against the PINsafe server.

Integrating with a user repository

PINsafe comes with an XML database that can be used to stored user accounts, however Enterprises may already have a user repository, eg Active Directory, for user accounts and this can be integrated with PINsafe.

The integration means that PINsafe will synchronise with the external user repository to ensure that the user accounts on PINsafe match those within the external user repository. To ensure that the two data stores remain synchronised PINsafe can be configured to synchronise with the external repository at regular intervals, eg once an hour, or synchronisations can be instigated manually.

How does the repository work?

PINsafe is configured so that administrators, helpdesk, users, single channel and dual channel transports are mapped to specific attributes and groups. These groups are created by the PINsafe administrator if using the XML repository or the AD Administrator for Active Directory.

Users are then simply added to the appropriate groups providing them with the permissions to use associated authentication method or rights.

This provides an extremely flexible mode of operation as different users can be grouped to use different transports or quickly configured to use an alternative transport or removed altogether.

How to integrate with Active Directory

Refer to Appendix B to see how Active Directory attributes relate to user repository groups.

Swivel provide a class for the integration of PINSafe with an Active Directory. This class acts as an interface between the PINsafe internal database and the external active directory database. This class can be set up from the Admin Console.

Go to the repository->general screen

1. Set an identifier for the Active Directory classs
2. Set the class name to com.swiveltechnologies.user.ActiveDirectory
3. Set the username attribute; this is the attribute within the Active Directory repository that will form the username within PINsafe
4. Optionally set the attributes within active directory that will form in the initial values for password and PIN

The Active Directory class now needs to be configured to point at the Active Directory server. To do this go to the Repository > Active directory screen.

1. Input the username. This username needs to be a fully qualified username for a user within the Active Directory domain that has the required privileges.
2. Input the password associated with the above account
3. The IP Address and required port number of the Active Directory Server.
4. If you are using SSL for the connection between PINsafe and Active Directory you need to select Port 636 and decide whether to accept self-signed certificates
5. Click apply.

The server is now configured to pull in user information from the Active Directory. There are two ways of instigated the pull of this data.

Manually	By going to the User Administration screen and selecting Sync.
Automatically	By going to the Server->Jobs screen you can set the interval (in seconds) between executions of the synchronisation process.

On the Repository -> General screen you can specify whether you import the disabled status of an AD account. If you choose this option any account that is disabled within active directory will be disabled in PINsafe. When an account is disabled it appears on the PINsafe server but the user cannot authenticate using that accoutn

Integrating with 3rd Party Authentication systems

PINsafe can be integrated with 3rd party authentication systems via its third party API. If this is required, a class needs to be developed that implements PINsafe third party API and interface to the third-party authentication system. The details of developing such as class are outside the scope of this document.

Assuming that this class exists, to use this class exists, PINSafe can be configured to use the third-party authentication system, via this class, for some or all users.

To configure PINsafe to do this you need to go to the Sever>Third Party Authentication screen.

Enter an identifier for the Authentication, the class, (that needs to be installed on the PINsafe server), and repository group that the use of the third party system will be associated with and any license key required by the third party system.

Once this configuration is in place the authentication process will be.

1. Agent submits authentication request, including credentials required by the third party system
2. The PINsafe server checks the user PINsafe credentials.
3. If this stage of authentication is successful and the user is a member of the repository group associated with the third party authentication class, the PINsafe server makes and authentication request to the third party system, passing the required credentials
4. The third party class returns a success code and, if this stage of authentication is also successful, successfully authenticates the user.

Integrating with Transports

A transport class is the mechanism by which a security string is delivered to the end user for dual factor authentication. The default method for this is via SMS and PINsafe comes with a class that enable this via a GSM Modem. However the transport class can be used to interface PINsafe with any suitable mechanism for carrying the string.

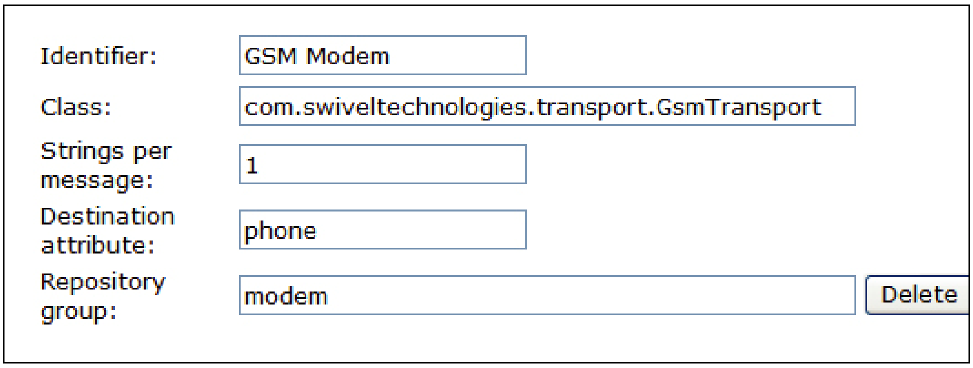
The API that a transport class needs to implement is very simple and consists of a single method.

How to integrate PINSafe with a GSM Modem

PINsafe supplies a transport class that allows PINsafe to send text messages via a GSM modem. (The GSM modem obviously needs a valid SIM card and GSM coverage). The transport class operates the modem by sending AT Modem commands via the COM1 serial port. Therefore, before configuring the PINsafe server, a AT Compatible modem needs to be connected to the COM1 serial port.

To configure the PINsafe server to use the modem.

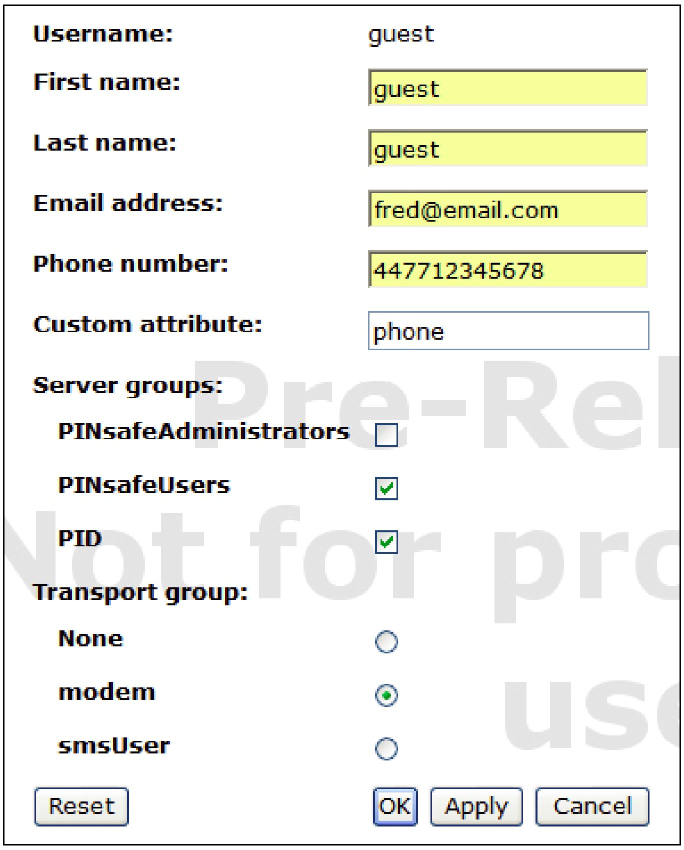
1. Go to the Transport > General
2. Ensure that the modem class details are entered on the screen as shown below
3. Enter a repository group name for the class, eg modem
4. Click apply.



Identifier:	GSM Modem
Class:	com.swiveltechnologies.transport.GsmTransport
Strings per message:	1
Destination attribute:	phone
Repository group:	modem

Transport class configuration screen

5. Once this entry has been made, the transport will appear in the left hand pane under the Transport heading. You can then configure the modem interface as required from the Transport > GSM Modem screen.
6. In order to configure users to use this transport to receive their security strings, user need to be members of the Modem user respority group. If you are using the XML repository you can affect this by going to the User Administration screen, selecting a user, and selecting modem as their transport group.



Username:	guest
First name:	guest
Last name:	guest
Email address:	fred@email.com
Phone number:	447712345678
Custom attribute:	phone
Server groups:	
PINsafeAdministrators	☐
PINsafeUsers	☑
PID	☑
Transport group:	
None	○
modem	●
smsUser	○

User configuration screen, user configured to use GSM modem

How to integrate PINsafe with an SMS service Provider

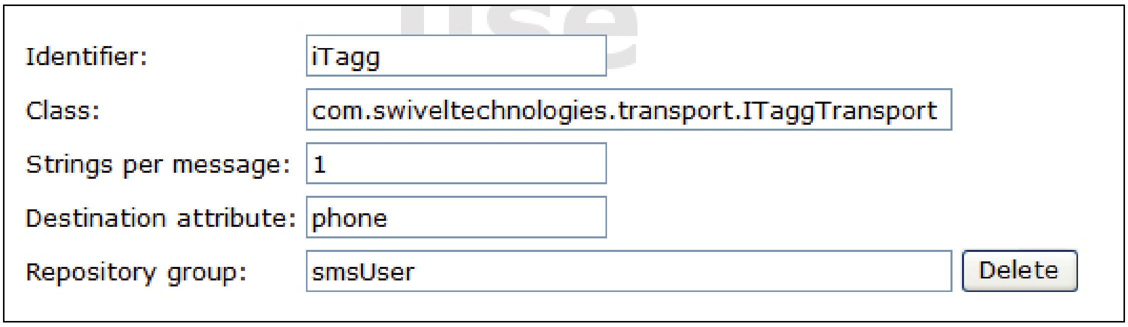
As an alternative to using a GSM Modem, PINSafe can be configured to use an SMS service provider. To do this you need to obtain an account with an SMS provider that can deliver SMS messages to the PINSafe user base. There are a number of such providers.

A transport class is required to act as an interface between PINsafe and the SMS provider; it receives messages from PINSafe and forwards them to the SMS provider in whatever format that the SMS provider requires.

Swivel has produced classes to work with a number of SMS providers including iTagg (www.itagg.com) and Clickatell (www.clickatell.com)

To configure PINsafe to use an SMS service provider.

1. Obtain an account from a SMS provider for which there is a transport class available.
2. Go to the Transport > General
3. Ensure that the SMS provider class details are entered on the screen as shown below
4. Enter a repository group name for the class, eg smsUser
5. Click apply.



Identifier:	iTagg
Class:	com.swiveltechnologies.transport.ITaggTransport
Strings per message:	1
Destination attribute:	phone
Repository group:	smsUser

Configuring an SMS provider transport class

6. Once this entry has been made, the transport identifier (eg iTagg) will appear in the left hand pane under the Transport heading. You can then configure this interface as required from the Transport > iTagg screen.
7. In order to configure users to use this transport to receive their security strings, user need to be members of the Modem user repository group. If you are using the XML repository you can affect this by going to the User Administration screen, selecting a user, and selecting smsUser as their transport group.

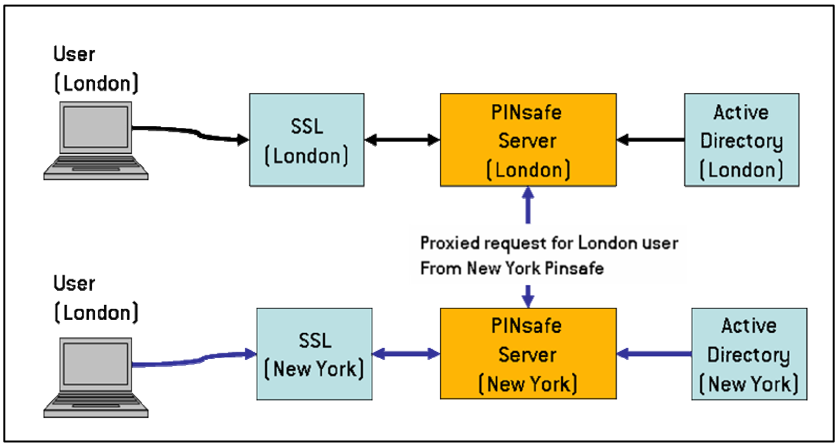
Peering PINsafe servers

It is now possible to deploy a number of PINsafe servers as a set of peers. Every user has an account on one of the PINsafe peers, however they can authenticate to any one of the PINsafe servers.

For example if a business has a London office and a New York office, running separate Active Directories and SSL VPNs a PINsafe server can be installed in each office. Each PINsafe server can be configured as a peer to the other.

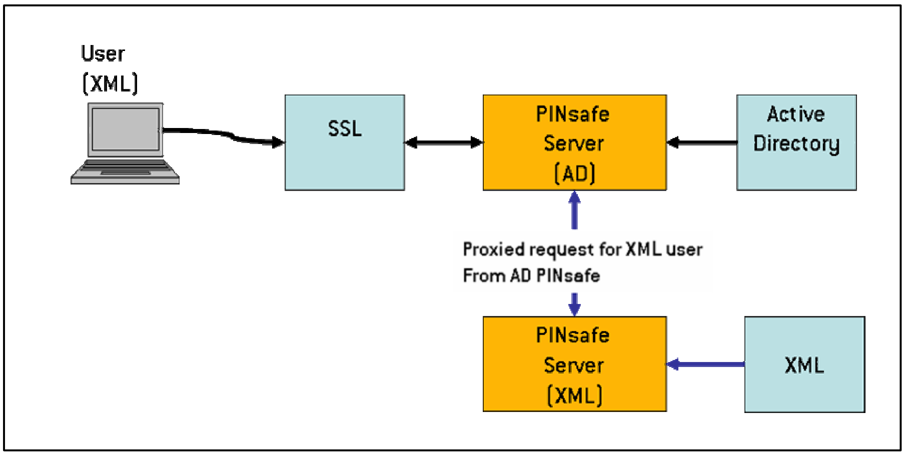
In this configuration, a London-based user can authenticate to the New York VPN; the New York PINsafe detects that the user is served by the London server and proxies the authentication request to the London PINsafe server.

The London PINsafe server checks the user's credentials and returns the results to the New York server that, as a result, can allow or deny access.



Example of peering for multi-office installation

Another application of this is to support the inter-working of different user repositories. One PINsafe server can be configured to work with Active Directory, another to work with the XML repository. The two servers can be configured as peers so that a user from either user repository can authenticate to a VPN (or web application) authenticating via either PINsafe server.

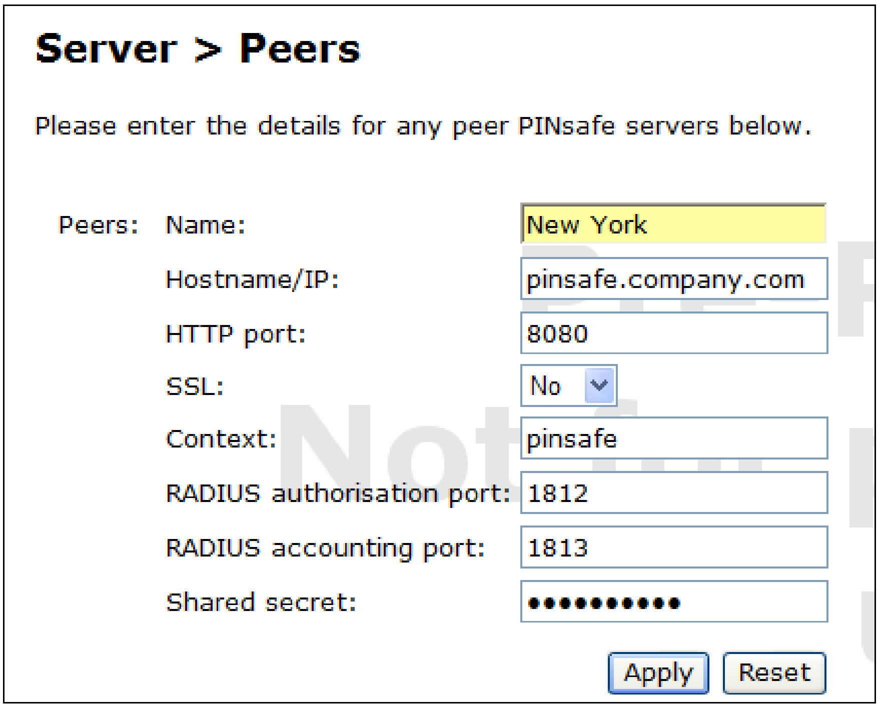


Example of peering fro multi-repository installation

It is possible to deploy both these servers on the same PINsafe appliance, thus avoiding increased hardware costs.

To implement peering each PINsafe Peer must know the IP address, context and port details of every other PINsafe server in the peering network. Each PINsafe server also has a shared secret used for authentication; to make authentication requests the requesting server must present the share secret to authenticate itself to its peer.

To set up this relationship enter the details on the Server-Peers screen on each PINsafe server. Using the London/New York example on the London PINsafe server the following would be entered.



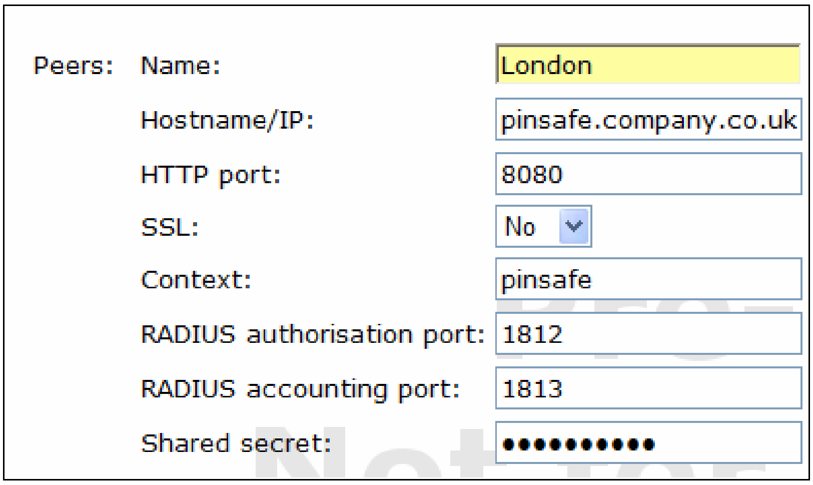
Server > Peers

Please enter the details for any peer PINsafe servers below.

Peers:	Name:	New York
	Hostname/IP:	pinsafe.company.com
	HTTP port:	8080
	SSL:	No
	Context:	pinsafe
	RADIUS authorisation port:	1812
	RADIUS accounting port:	1813
	Shared secret:	••••••••••

Example Peer configuration screen

And then on the New York server would be entered



Peers:	Name:	London
	Hostname/IP:	pinsafe.company.co.uk
	HTTP port:	8080
	SSL:	No
	Context:	pinsafe
	RADIUS authorisation port:	1812
	RADIUS accounting port:	1813
	Shared secret:	●●●●●●●●●●●

Example Peer configuration screen

Peering can be used for RADIUS and agent XML authentication solutions. An inbound RADIUS request will be proxied via RADIUS to a peer PINsafe server as required. Similarly an Agent-XML based authentication request will be proxied via the Agent-XML interface.

Peering works by each peer keeping a record of the user-names active on the other servers. This list is updated periodically. How frequently and when this synchronisation takes place is configured on the server>jobs page; see section on Jobs.

Operation & Maintenance

How to Log on to the PINsafe server

The PINsafe server is delivered with a default root password of lockbox. It is recommended that this is changed (and, obviously, recorded) as part of the installation process.

How to perform back-ups of the PINsafe server

To back up the user and configuration data for the server, the easiest approach is to back up the webapps under the Tomcat directory.

usr/local/jakarta-tomcat-x.x-xx/webapps

Where x.x-xx is the tomcat version number.

This will back up all the user and system data required by PINsafe.

To ensure against sever hardware failure and to facilitate quick restore it is recommended that this back up should be stored on a remote server.

Saving the configuration

PINsafe has a save configuration feature that allows the current configuration to be saved to an xml file. This provides a useful record of the PINsafe configuration and can be useful for fault diagnosis

How to perform automated backups

The following guidelines are a suggested way of performing backups automatically in way that can be integrated with any existing enterprise back-up processes and sytems.

This approach uses a back up script to copy the required files to a SAMBA share that allows the back-ups to be copied via the local network to remote server.

Creating a Samba Share:

This is a very quick guide to creating a public share using SAMBA, no security policies have been added to the share. It is being used as a form of access from your global backup server.

Firstly at the console, log in as root.

Start the x-display by typing *startx*

Once the graphical interface has started select *Applications – System Settings – Server Settings – Samba*

Select *Add*

In the Directory browse to and create a new folder called *PINsafeBackup* in the */home/swivel* directory. Then select the new folder as the samba share directory.

N.B. The backup script provided later presumes a /home/swivel/PINsafeBackup folder exists, if you change this folder here remember to change the backup script to match

Select *Read* and *Allow access to everyone*

In the *Preferences – Server Settings - Security* menu item set the Authentication Mode to *Share*

In the *Applications – System Settings – Server Settings – Services* menu item tick the *smb* service box and also select *Start Service*

You should now be able to access the share via a client machine pointing to //<server_ip>/PINsafeBackup where <server_ip> is the PINsafe server's IP address.

Writing a simple script to back up PINSafe:

Log onto the console as root.

Use an editor to create a file */home/swivel/backup.sh*

Enter the following code **nb** replace the tomcat-x.x-xx version with the correct version number running on your server.

```
#! /bin/bash
/etc/init.d/tomcat stop
cd /usr/local/jakarta-tomcat-x.x-xx/webapps
tar --create \
--file=/home/swivel/PINsafeBackup/Backup.tar \
--label=`Backup` \
--verbose \
pinsafe
/etc/init.d/tomcat start
```

To make the script executable type:

```
chmod a+x /home/swivel/backup.sh
```

Running this script temporarily halts the PINsafe server and therefore prevents users authenticating via PINsafe during back-ups.

Running the script

To run the script manually do the following:

```
/home/swivel/backup.sh
```

There should appear a new tar file called Backup.tar in your SAMBA share.

How often should I perform back-ups ?

It is recommended that this script be run after the first user repository synchronization and then on a regular (eg weekly) basis. How regularly it is run, depends on how often users and PINs changes.

The script can be run as a cron job. The used of a SAMBA share means that this data can then be pulled from the machine and stored elsewhere to conform to any back-up routines/infrastructure that may already be in place.

These instructions are only a guideline and we would recommend you further your investigations into backup.

Restoring from Backup

If you should need to restore your server from a backup, carry out the following:

Log into the server as root and stop Tomcat:

```
/etc/init.d/tomcat stop
```

Delete the pinsafe folder from within the tomcat webapps directory

```
cd /usr/local/jakarta-tomcat-x.x-xx/webapps

rm –rf pinsafe
```

Copy and extract the backup file

```
cp /home/swivel/PINsafeBackup/Backup.tar /usr/local/jakarta-tomcat-x.x-xx/webapps

tar –xvf /usr/local/jakarta-tomcat-x.x-xx/webapps
Backup.tar
```

Restart Tomcat and check PINsafe is running and your settings are correct.

```
/etc/init.d/tomcat start
```

In a browser point to the admin console: http://server_ip:8080/pinsafe.

How to perform disaster recovery

In the event of a hardware failure or other scenario that requires a new server to be installed and brought up to the last recorded configuration of the live server.

1. Install PINsafe and associated software from disks supplied as part of the install
2. Restoring copy the latest back-up of the web apps folder to web apps
3. Restarting the tomcat server

Jobs

There are a number of processes (or jobs) that the server needs to run on a regular basis. These handle such things as synchronizing to the user repository and checking for any accounts that should be locked due to inactivity.

For the most part these settings can be left to their default values but there maybe reasons why an administrator would want to change these settings. When choosing these settings the administrator needs to balance the requirement to synchronize data regularly and the resultant loading on the server. Where possible these tasks should be scheduled to run during the server's quite period.

Session Clean Up

The session clean-up job is used to invalidate, after a given time, any security strings that have been requested by the user.

For example a session is deemed to have started when a TURing image is requested. The security string presented within that security string is only valid for as long as the session is valid. The length of time for which the session is valid is set by the session clean-up time; if it is set to 120 seconds the user will have 2 minutes in which to use the security string to authenticate. If they attempt to authenticate with that security string after that time their authentication will fail.

The same setting also applies to security strings delivered by SMS messages when they have been explicitly requested by the user, eg via a GET MESSAGE button. SMS messages sent automatically, after an authentication attempt, do not expire in this way.

Scheduler settings

With the exception of the session clean up setting, tasks are scheduled using a *cron* syntax.

The use of the cron-like syntax gives a great deal of flexibility in scheduling these tasks. The settings require the following fields.

Field Name	**Allowed Values**	**Allowed Special Characters**
Seconds	0-59	, - * /
Minutes	0-59	, - * /
Hours	0-23	, - * /
Day-of-month	1-31	, - * ? / L W C
Month	1-12 or JAN-DEC	, - * /
Day-of-Week	1-7 SUN-SAT	- * ? / L C #
Year (Optional)	empty, 1970-2099	, - * /

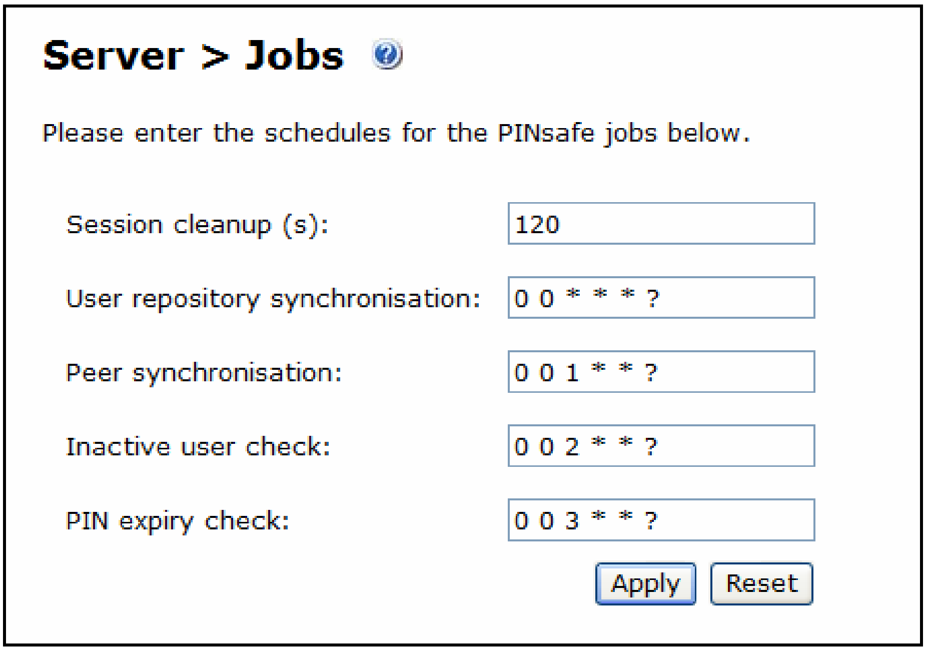
These fields determine when the job is to be run. An asterisk (*) in any field is a wildcard and means that the task will be run for all values of that field. A question-mark (?) means that the setting is not specified, this setting is applied to either the day-of-month and day-of-week as you cannot specify both of these parameters.

Therefore to schedule a job to run every hour, on the hour the settings would be

0 0 * * * ?

Whereas to run a job 3am every Sunday would be

0 0 3 ? * 1



Server > Jobs

Please enter the schedules for the PINsafe jobs below.

Session cleanup (s):	120
User repository synchronisation:	0 0 * * * ?
Peer synchronisation:	0 0 1 * * ?
Inactive user check:	0 0 2 * * ?
PIN expiry check:	0 0 3 * * ?

Server > Jobs scheduling Screen; User repository job running every hour, Peer Synchronisation, Inactive user check and PIN expiry check running daily at 1am, 2am and 3am respectively

User Repository

This job synchronizes the user repository, eg Active Directory, with the PINsafe server. The more users in the Active Directory the longer this job is liable to take. If you add a PINsafe user to the active directory that user will not be active on PINsafe until this job has run or until an administrator has manually instigated the synchronization from the user admin screen.

Peer Synchronization

This is similar to the user repository synchronization only it refers to a server synchronizing its list of users with other peer servers within its peer network. Every time this job is run, the server will request an up to date user list from its peer servers.

Inactive User Check

If there are policies in place on the server to lock accounts that have been inactive for more than a certain time then this job will detect those inactive accounts and lock them.

PIN expiry check
If there are policies in place on the server limit how long a PIN is valid for, this job will go through the user list and check to see the last time the pin was changed and then either do nothing, send out a PIN expiry warning or lock out the account.

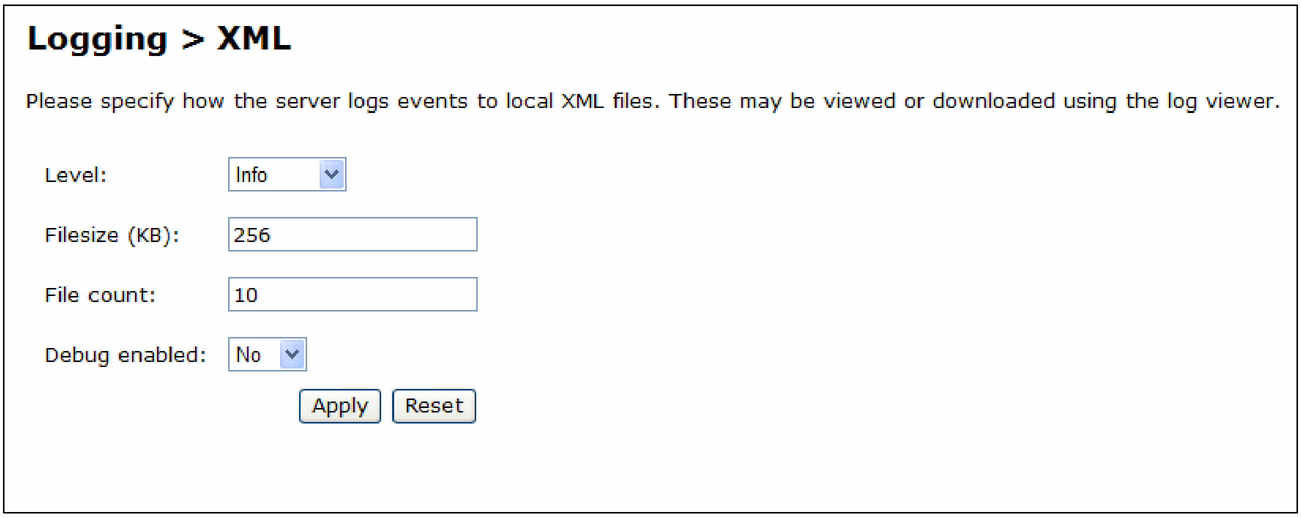
Logs and Alarms

PINsafe generates a range of log events and alarms they consist of the following options.

- XML Log files written to the local PINsafe server
- Log files written to Syslog
- Alarm events sent as emails to a specified email address.

XML Logging

XML logs are generated for a number of system events; they have varying levels of severity. FATAL, ERROR, WARNING, INFO. You can set the level of logging; setting the level to INFO will mean all events of severity INFO and above will be recorded.



Logging > XML

Please specify how the server logs events to local XML files. These may be viewed or downloaded using the log viewer.

Level: Info

Filesize (KB): 256

File count: 10

Debug enabled: No

Screen for configuring XML Logging

The log events are written to files on the PINsafe server at <tomcat>\webapps\pinsafe\WEB-INF\logs.

If you are running a backup script like the one described earlier in this manual then these log files can be included in that back up to provide a longer term log of system activity.

A log file is written (pinsafe.log) to until it reaches the file size specified on the Logging > XML screen. This file is then renamed to pinsafe.log.1, and a new pinsafe.log is created and writing resumes to that file. This process repeats, creating pinsage.log.2, pinsafe.log.3 etc.

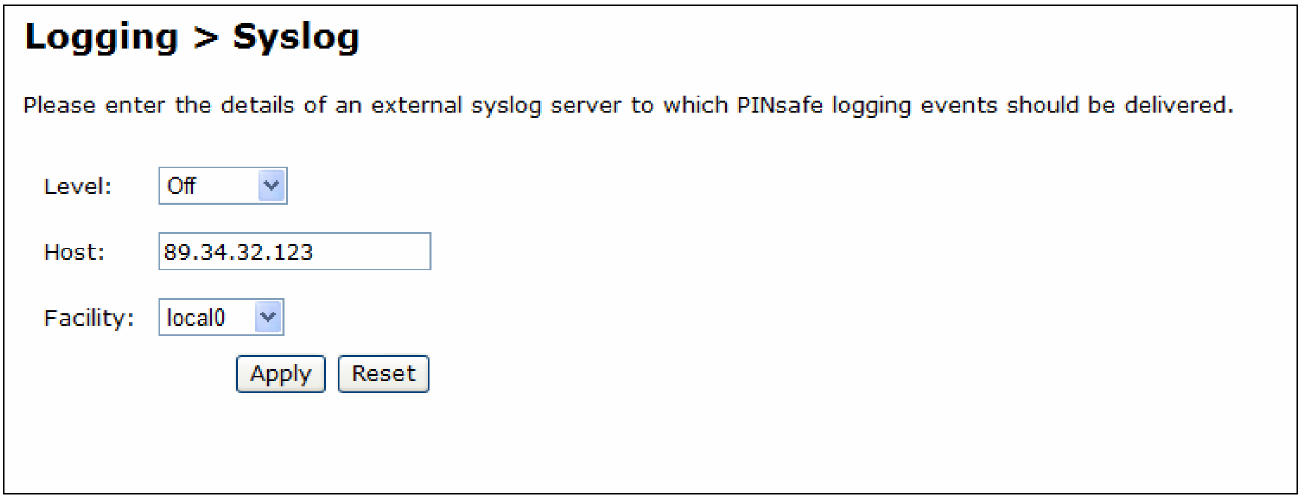
The number of log files used is determined by the File Count entry. Once this count is reached, the oldest log file on the server is overwritten.

If debug is enabled, debug logs are created that give much more detailed information about the processes running within the server. This setting creates large log files and has and impact on the performance on the server and therefore should only be used for fault diagnosis. Debug logs are written to a separate file, <tomcat>\webapps\pinsafe\WEB-INF\logg\debug.log

The contents of the XML log files can be viewed via the PINsafe Administration interface, Log Viewer screen and can be downloaded from the PINsafe server to a local machine.

Syslog

As an alternative or addition to writing XML log files locally, PINsafe can also write log files remotely by using the Syslog logging feature.



Logging > Syslog

Please enter the details of an external syslog server to which PINsafe logging events should be delivered.

Level: Off

Host: 89.34.32.123

Facility: local0

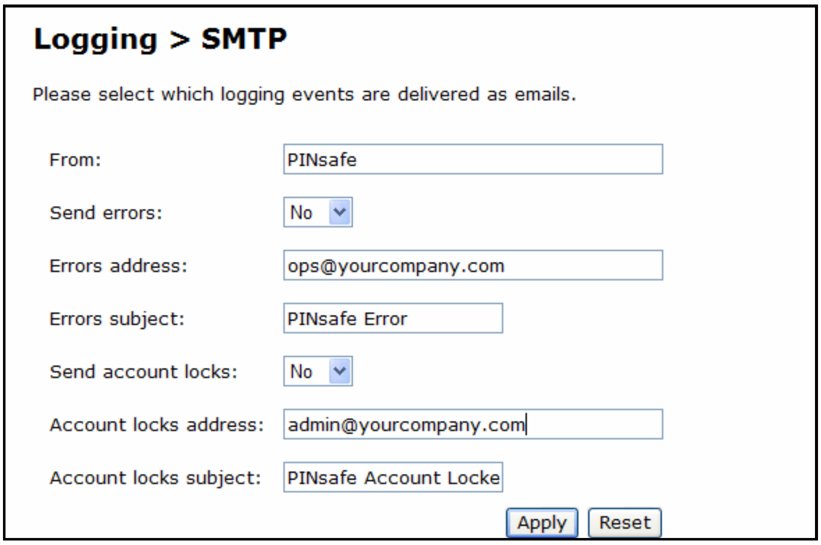
Syslog configuration screen

The logging level is set in the same was as for XML logging. The additional information required for Syslog is the host to which the logs will be written and the syslog facility to be used.

SMTP (email) Logging

Certain events can me emailed to operational managers. These are system errors and account lock-outs. The configuration screen for this feature is shown below.

To use SMTP logging you must have access to a suitable SMTP mail- server. The details for this server must be entered on to the server > SMTP screen.



Please select which logging events are delivered as emails.

From:	PINsafe
Send errors:	No
Errors address:	ops@yourcompany.com
Errors subject:	PINsafe Error
Send account locks:	No
Account locks address:	admin@yourcompany.com
Account locks subject:	PINsafe Account Locke

SMTP Logging configuration screen

Administrator's Guide

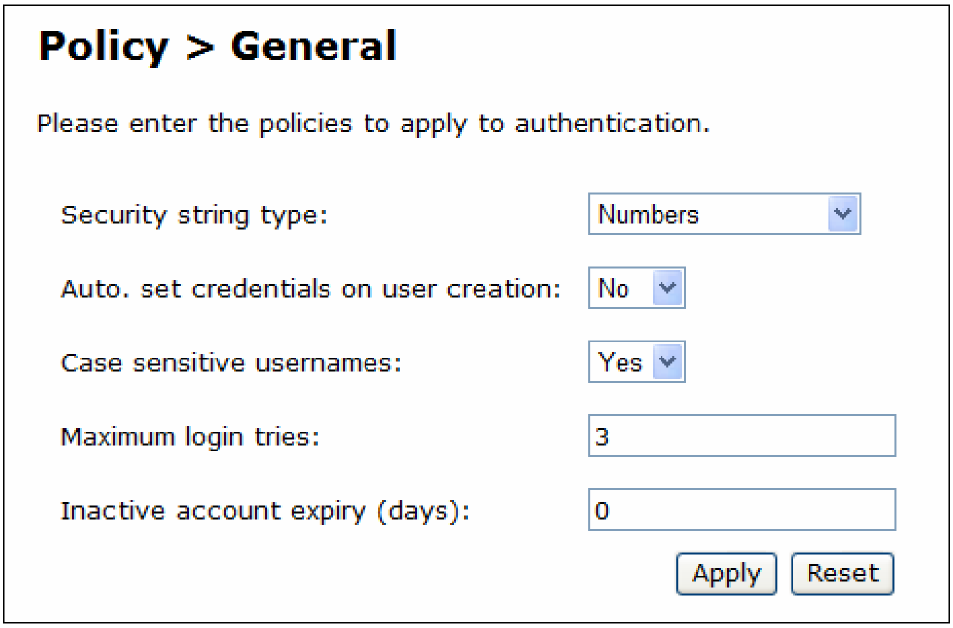
This section concentrates on explaining how the PINsafe server is administered; it covers all the common tasks that an administrator would normally undertake. **Nb** A full online help admin reference guide is provided as part of the Swivel distribution.

It assumes all the required integration tasks described earlier in this document have been completed.

Setting Policies

There are a number of policies relating to account/PIN management, for example PIN length and PIN validity periods.

The general policies are set on the Policy > General screen.



Policy > General

Please enter the policies to apply to authentication.

Security string type:	Numbers
Auto. set credentials on user creation:	No
Case sensitive usernames:	Yes
Maximum login tries:	3
Inactive account expiry (days):	0

General Policy Screen

Security String Type

You can set the security string to be Numbers, Letters. You can have upper or lower case characters or a mixture of both. Mixed case is not recommended for TURing images as it can be difficult to differentiate between characters such as lower case l and number 1, even without the obfuscation.

If the SMS delivery of security strings is used; then it is recommended that numbers are used for security strings.

User Name case sensitivity

You can select whether usernames are case sensitive or not. You may need to set usernames to be case insensitive if that is what the users are used to.

Maximum login tries

This is the maximum number of consecutive failed login attempts that a user can have before their account is locked out. see account unlocking accounts.

Inactive account expiry

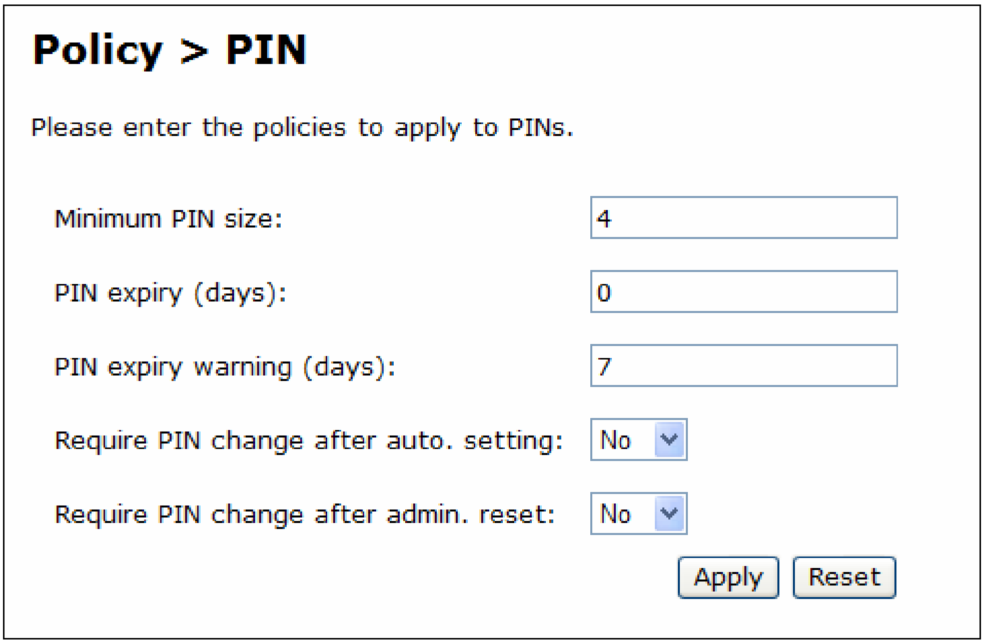
If an account is not used for specified period, it will become locked. This setting allows you to specify that period (in days). If this is set to 0 then the user account never expires due to inactivity.

Auto Credential setting

If you set this parameter to YES, then whenever a new account is created, PINsafe will automatically create a PIN for them. If a password is also required, this will also be generated, see *Adding Users* for more details.

PIN Policies

PIN policies are set on the Policy > PIN screen. PINs within the PINsafe product are the prime authentication credential and administrators may wish to replicate existing password management policies within PINsafe.



Policy > PIN

Please enter the policies to apply to PINs.

Minimum PIN size: 4

PIN expiry (days): 0

PIN expiry warning (days): 7

Require PIN change after auto. setting: No

Require PIN change after admin. reset: No

PIN Policy Screen

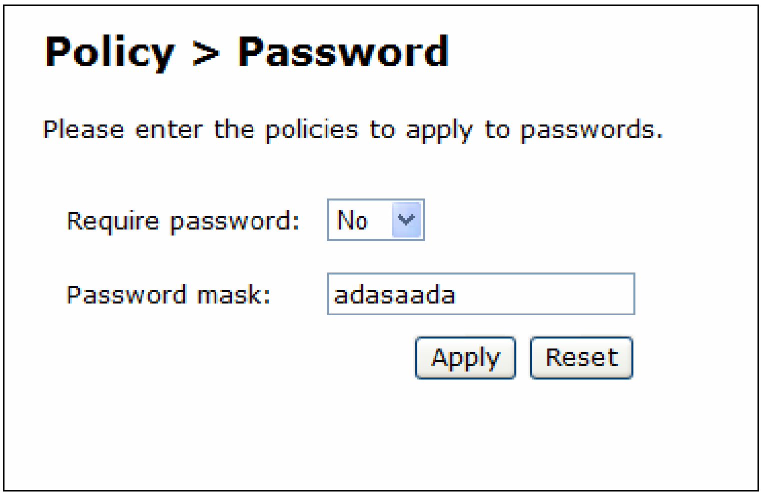
From this screen you can set the minimum PIN size (from 4 to 10 characters). You can also set a PIN expiry period. This determines how long a PIN is valid for, for example setting this figure to 90 days will ensure that users will need to change their PIN at least every 90 days. Setting this value to 0 (zero) would mean that the PINs would never expire.

You can ensure that users are warned about their impending PIN expiry. Setting the PIN expiry warning will determine how many days in advance the users will be prompted to change their PIN. The content and delivery of the warning will depend on which alert group the user is a member.

Other PIN policies that PINsafe can implement is to require a PIN change after the user has had a PIN auto-created (eg via auto credential creation) or where a user has had their PIN change by the Administrator.

Where these policies are enforce, a user must authenticate and then change their PIN; the PINs that have been created for them will only work once.

Password Policies



Policy > Password

Please enter the policies to apply to passwords.

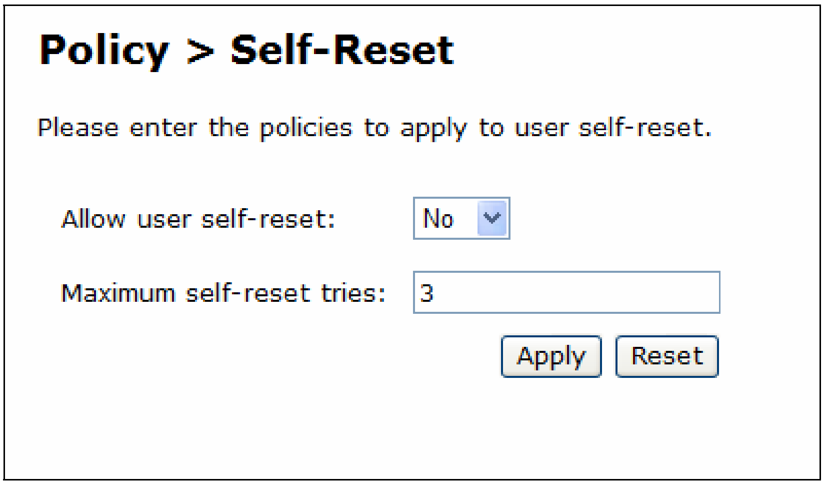
Require password: No

Password mask: adasaada

Password Policy Screen

Passwords are an optional part of the PINsafe authentication model. If required the Administrator can ensure that all accounts have a password as well as a PIN associated with them. If passwords are required then if auto-credential creation is set, PINsafe will create a random password for the user. The random password will conform to the password mask. The password mask allows administrators to ensure that certain character types are included in the password in the specified order where: a = alpha, d = digit and s = special character. An example of a password conforming to the above password mask would be r4p&dl2a.

User self-reset



Policy > Self-Reset

Please enter the policies to apply to user self-reset.

Allow user self-reset: No

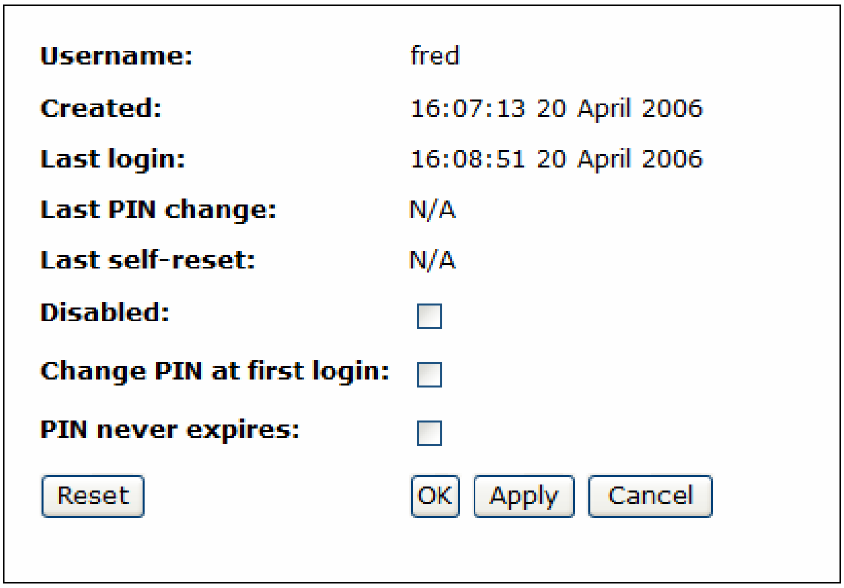
Maximum self-reset tries: 3

Self Reset Policy Screen

PINsafe supports a self-reset policy, whereby if a user's account has been locked they can unlock it. They do this by being sent an unlock code via their alert transport, they then enter this code to authenticate. The above screen enables this feature and stipulates the maximum number reset tries a user is allowed.

User Policies

Along with the server-wide policies the Administrator can set policies for individual users. To access this feature the Administrator can go to the User Administration screen and select a user and then select the Policy button. This will bring up the following screen for that user.



Username:	fred
Created:	16:07:13 20 April 2006
Last login:	16:08:51 20 April 2006
Last PIN change:	N/A
Last self-reset:	N/A
Disabled:	□
Change PIN at first login:	□
PIN never expires:	□

User Policy Screen

This screen shows the status of the user, when they were created on the system and when the last logged-in etc. The Administrator can also, from this screen, implement the following policies.

Disabled: If an account is disabled a user cannot authenticate. Accounts can only be enabled again from this screen. Administrators may wish to disable accounts when a user no longer requires access but they wish to retain the information associated with that account.

Change PIN: The Administrator can ensure that the a user has to change their PIN at their next login; ie their PIN will only be valid for one authentication

Never Expires: The PIN for this account will never expire, this takes precedent over any server-wide PIN policy. Administrators may wish to use this feature for admin accounts.

Single Channel

The Servers > Single channel screen allows the Administrator to specify how single channel security strings are presented to the user. From this screen the you can specify whether the security string image will be displayed as a TURing, PATTern or BUTTon image and whether the characters will be rotated within the image. In this screen you can also determine whether to allow session creation via username. This means allowing a user to request a TURing image from any url for example by pasting http://<pinsage url>/pinsafe/SCImage?username=<username> into a web browser.

This provides flexibility in terms on security string delivery but does require that port 80 is opened up on the PINsafe server.

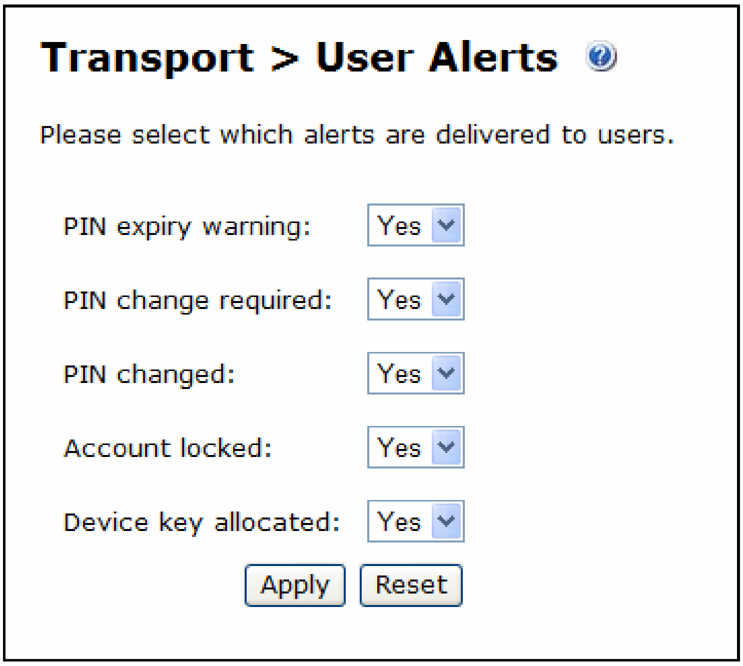
Dual Channel

The server > dual channel screen determines how dual channel security strings are delivered. The default model for dual channel security strings is that a new security string is sent whenever a user attempts to authenticate, thus ensuring that the user always has a valid security string to use.
An alternative is to only send a security string to the user when they explicitly request one; this is the on-demand mode that can be enabled from this screen. If this mode is used the security string sent to the user is only valid for as long as the session timeout (clean-up) period specified on the server > logs screen.

The request to send a dual channel security string can be via an agent configured on the PINsafe server; alternatively PINsafe can be configured to allow the request for a security string to be instigated from any IP address in a similar way to requesting TURing images. This is enabled by setting Allow message request by username to Yes

Alerting Users

The Transport > Alerts screen can be used to determine for which events users receive alerts. Alerts will be sent to the user based on which Alert Transport group to which they belong.



Transport > User Alerts

Please select which alerts are delivered to users.

PIN expiry warning: Yes

PIN change required: Yes

PIN changed: Yes

Account locked: Yes

Device key allocated: Yes

Alert Configuration Screen

Managing Users

This section covers the management of users within the PINsafe server; including the addition and removal of users, unlocking accounts.

Adding Users

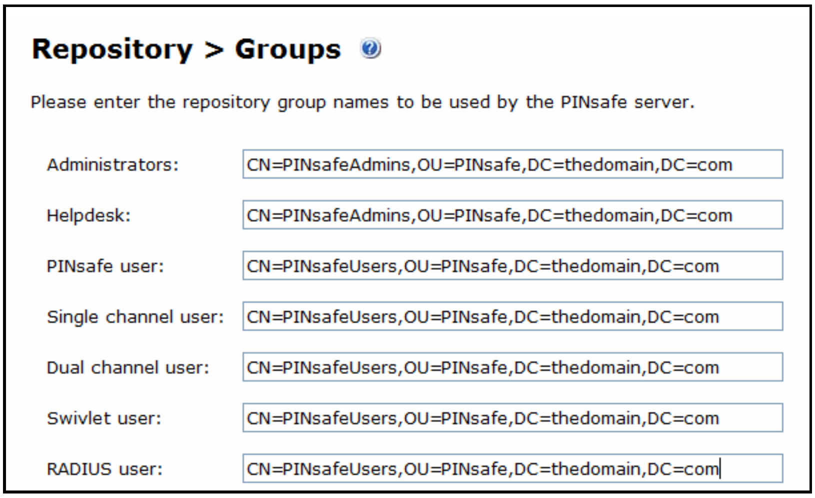
When adding users to the PINsafe server you need to consider the following:-

- What rights will they have, eg will they be able to use single channel, dual channel or both
- How will security strings be delivered to them
- How will their PIN (and Password) be created and delivered to them.

The implementation of these will be achieved by ensuring that the user is a member of the associate user group.

Allocating users to authentication methods

If the user accounts are being synchronized with Active Directory the user must be a member of the correct groups with Active Directory to be able to use the associated authentication methods within PINsafe. The Repository > Groups screen will show of which Active Directory groups the user must be a member. Different authentication options may be associated with different groups within Active Directory or, as in the example below, all users will have access to all authentication types.



Repository > Groups

Please enter the repository group names to be used by the PINsafe server.

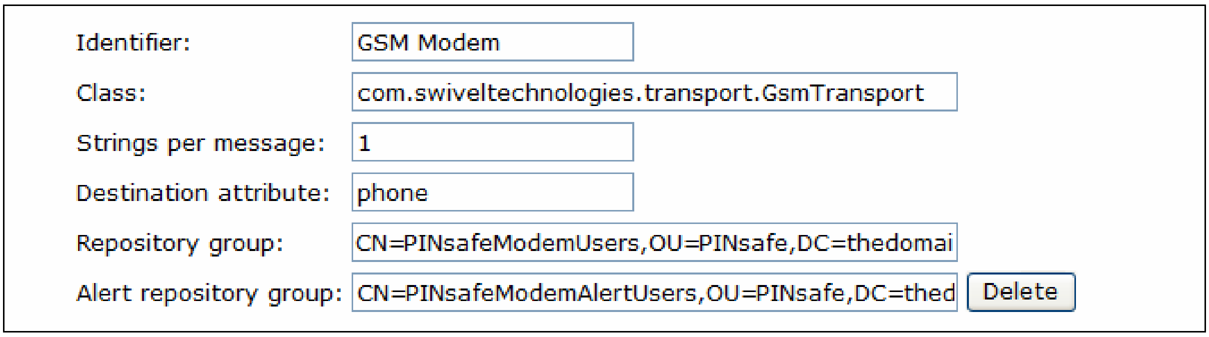
Administrators:	CN=PINsafeAdmins,OU=PINsafe,DC=thedomain,DC=com
Helpdesk:	CN=PINsafeAdmins,OU=PINsafe,DC=thedomain,DC=com
PINsafe user:	CN=PINsafeUsers,OU=PINsafe,DC=thedomain,DC=com
Single channel user:	CN=PINsafeUsers,OU=PINsafe,DC=thedomain,DC=com
Dual channel user:	CN=PINsafeUsers,OU=PINsafe,DC=thedomain,DC=com
Swivlet user:	CN=PINsafeUsers,OU=PINsafe,DC=thedomain,DC=com
RADIUS user:	CN=PINsafeUsers,OU=PINsafe,DC=thedomain,DC=com

Simple Active Directory Group Structure

Allocating Users to Transports

Users need to be associated with transport classes for the delivery of security strings and for system alerts, eg notification of PIN.

The Transport > General screen shows which groups are associated with which transports. It also shows the destination attribute therefore the Administrator needs to ensure that this attribute is set for the user.



Identifier:	GSM Modem
Class:	com.swiveltechnologies.transport.GsmTransport
Strings per message:	1
Destination attribute:	phone
Repository group:	CN=PINsafeModemUsers,OU=PINsafe,DC=thedomai
Alert repository group:	CN=PINsafeModemAlertUsers,OU=PINsafe,DC=thed

Screen showing AD groups associated with GSM Modem transport.

The Administrator may wish to use different transports for alerts from security string delivery.

Creating the account

Once the user has been made a member of the relevant groups the repository can be synchronised with PINsafe; this can either be done manually via the User Admin screen or automatically be setting up a job to perform this. (Server-Jobs, see Operation & Maintenance)

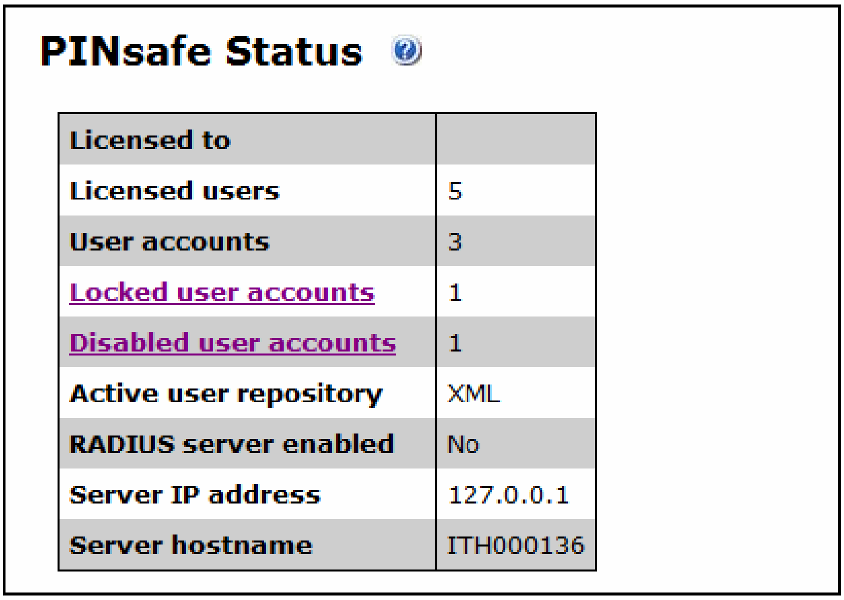
PINsafe will create a new account; if auto-credentials create is enabled, PINsafe will also create a username (and password) if required and send them to the user via their allocated Alert transport.

Unlocking Accounts

Accounts can be locked or disabled. If a user's account is locked or disabled they will not be able to authenticate via PINsafe.

You can configure PINsafe to send an email to an administrator to inform them when an account has become locked. See Logs and Alarms.

The main status screen on PINsafe immediately indicates if there are locked or disabled accounts on the PINsafe server.



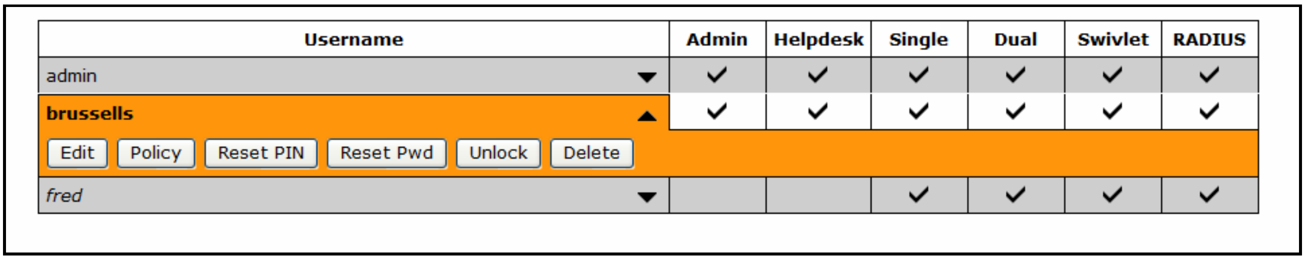
PINsafe Status

Licensed to	
Licensed users	5
User accounts	3
Locked user accounts	1
Disabled user accounts	1
Active user repository	XML
RADIUS server enabled	No
Server IP address	127.0.0.1
Server hostname	ITH000136

PINsafe Status Screen

The entries for the Locked and Disabled are hyperlinks to the user administration page; clicking on these links will take you to a list of all Locked accounts and all Disabled accounts respectively.

One the User Administration screen, locked accounts are shown in **bold** and disabled accounts are in *italics.*



Username	Admin	Helpdesk	Single	Dual	Swivlet	RADIUS
admin	✓	✓	✓	✓	✓	✓
brussells	✓	✓	✓	✓	✓	✓
fred			✓	✓	✓	✓

Accounts listed on User Administration screen

To unlock an account, click on the account in question and then on Unlock. To enable a disabled account, click on the Policy button and uncheck the disabled checkbox.

Appendix A Windows Installation of the Java Comm API

1. unzip the file javacomm20-win32.zip into the root of C:

This will produce a hierarchy with a top level directory commapi.

This example assumes that you have installed the J2SE version 5 and has been installed into C:\Program Files\Java\jdk1.5.0_02

2. Copy win32com.dll to your Java\jdk....\jre\bin directory.

```
C:\>copy c:\commapi\win32com.dll to C:\Program Files\Java\jdk1.5.0_02\jre\bin
```

3. Copy comm.jar to your Java\jdk....\jre\lib\ext directory.

```
C:\>copy c:\commapi\comm.jar to C:\Program Files\Java\jdk1.5.0_02\jre\lib\ext
```

4. Copy javax.comm.properties to your Java\jdk....\jre\lib directory.

```
C:\>copy c:\commapi\javax.comm.properties to C:\Program Files\Java\jdk1.5.0_02\jre\lib
```

5. Go to Environment Variable and add the Java\jdkƐ.\bin to ‘path’

```
 C:\Program Files\Java\jdk1.5.0_02\bin
```

TESTING

The Comm API also comes with a number of samples that can be used to confirm the Comm API has been installed correctly.

One of these samples is called BlackBox.

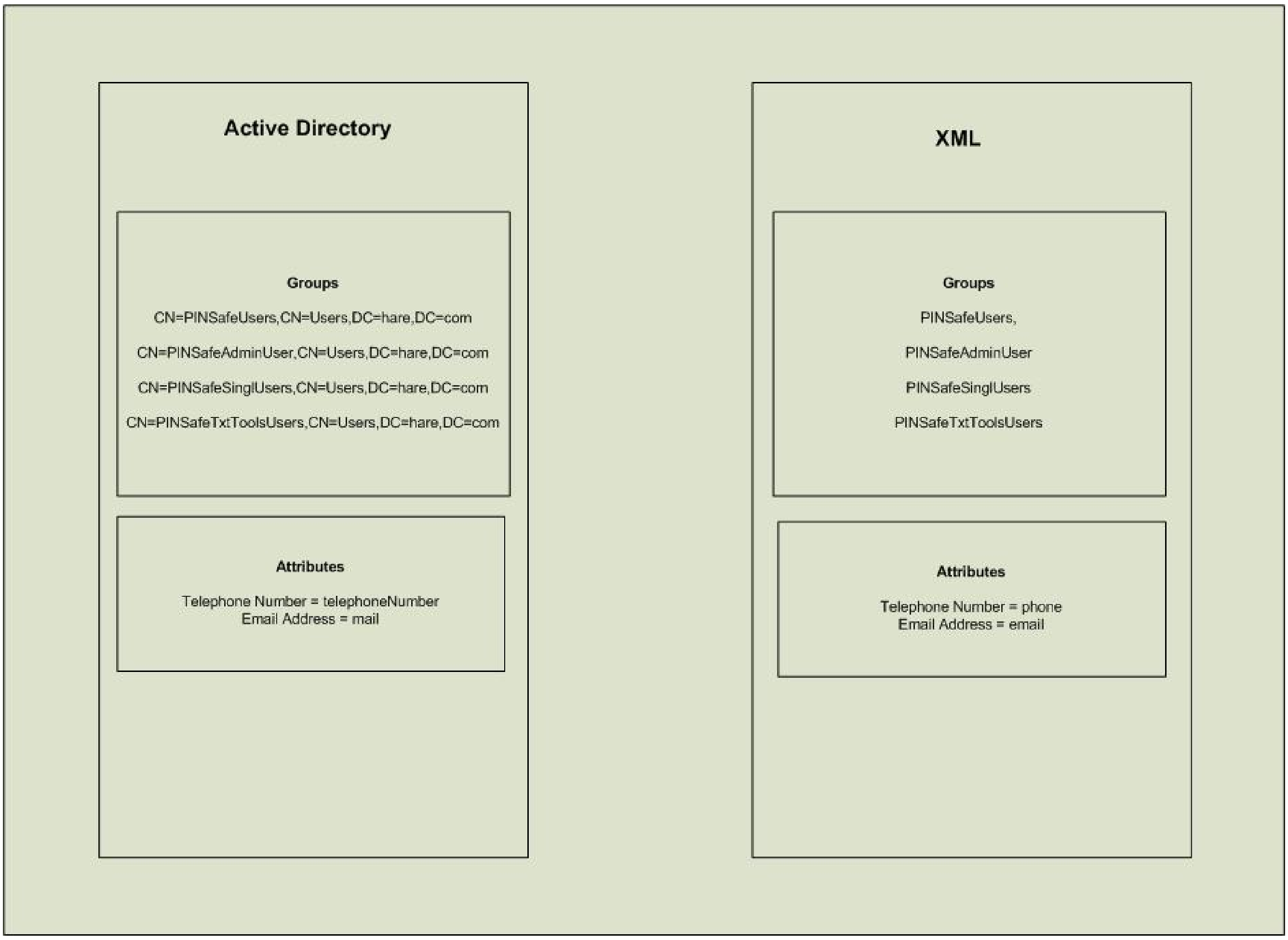
To use Blackbox add BlackBox.jar to the CLASSPATH in Environment Variables

```
        C:\commapi\samples\BlackBox\BlackBox.jar
```

To run BlackBox, open a command prompt and go to C:\commapi\samples\BlackBox. Then enter 'java BlackBox'

```
        C:\commapi\samples\BlackBox>java BlackBox
```

Appendix B An example of AD and XML groups and attributes


Active Directory
Groups
CN=PINSafeUsers,CN=Users,DC=hare,DC=com
CN=PINSafeAdminUser,CN=Users,DC=hare,DC=com
CN=PINSafeSinglUsers,CN=Users,DC=hare,DC=com
CN=PINSafeTxtToolsUsers,CN=Users,DC=hare,DC=com
Attributes
Telephone Number = telephoneNumber
Email Address = mail
XML
Groups
PINSafeUsers,
PINSafeAdminUser
PINSafeSinglUsers
PINSafeTxtToolsUsers
Attributes
Telephone Number = phone
Email Address = email

Appendix C PINsafe Installation details.

General	
Date of Install	
Installation company name	
Engineer's name	
Customer Contact (inc email/phone)	

PINsafe Server Details	
Appliance	Yes/No
Appliance serial number(s)	
High Availability	Yes/No
License Installed	Yes/No
PINsafe version	
OS	
IP Address	
Default gateway	
DNS	
Language	English
RADIUS	Yes/No
Change PIN	Yes/No
User self reset.	Yes/No
PINsafe Backup details	

Repository Details	
Repository	XML/Active Directory
AD IP Address	
AD Idap account name	@
Administrator group name	
Helpdesk group name	
Single channel user group name	

Dual channel user group name	
Swivlet user group name	

Authentication Details	
Single channel	Yes/No
Single channel option	
Dual Channel	Yes/No
Transports configured	
GSM Modem installed	Yes/No
SIM provider	
Alerts	Yes/No
Alerting transport	

Summary of solution.

Agent Configuration	
Authentication Agent(s)	
Agents Details	

Radius Configuration	
NAS	
PINsafe NAS details	

Support Arrangements	
Hardware	
OS	
Software	

Customer Sign-off

The installation has been completed to my satisfaction in accordance with the details given above.

I understand the level of support I am entitled to as explained above.

Signature________________ Date: ______

Name___________________ on Behalf of (company) __________________ ______________

Index