

Reporting How to guide

Contents

- 1 Overview
- 2 Prerequisites
- 3 Reporting access and restrictions
- 4 Producing Instant Reports
- 5 Producing Scheduled Reports
 - ◆ 5.1 Scheduled report file locations
- 6 Report date Format
- 7 Available Reports
- 8 Custom Reports
 - ◆ 8.1 Adding new reports
 - ◇ 8.1.1 Creation time, last login and number of logins
 - ◇ 8.1.2 List all users, together with their login count
 - ◇ 8.1.3 List users of a particular group showing their last login over the last month
 - ◇ 8.1.4 List Inactive Users Within a Period By Group
 - ◇ 8.1.5 Total number of users
 - ◇ 8.1.6 Total number of users in each group
 - ◇ 8.1.7 List users in each group
 - ◇ 8.1.8 List all OATH users
 - ◇ 8.1.9 List users with Mobile permissions
 - 8.1.9.1 List users with a provisioned Mobile device
 - 8.1.9.2 List users with Mobile client rights that have not yet provisioned a device
 - 8.1.9.3 List the number of login failures for all users
 - 8.1.9.4 List of users that provisioned before a given date
 - 8.1.9.5 List unprovisioned Mobile Users for a given date
 - ◇ 8.1.10 List the number of login failures since a given date, for all users
 - ◇ 8.1.11 List the last PIN change for all users
 - ◇ 8.1.12 List PIN changes older than a given date
 - ◇ 8.1.13 Lists the number of logins for each hour
 - ◇ 8.1.14 List number of logins for each day and hour for the last 7 days
 - ◇ 8.1.15 Enhancement to the previous report to include login failures and restrict to a single group
 - ◇ 8.1.16 List users that have been recently deleted
 - ◆ 8.2 Internal database reports
 - ◇ 8.2.1 Internal database, lists users that have not logged in since a specified date
 - ◆ 8.3 Custom reports for multiple database types
 - ◇ 8.3.1 Users marked as deleted
 - ◇ 8.3.2 Users whose PIN never expires
 - ◇ 8.3.3 All User Logins, including login location
 - ◇ 8.3.4 All user logins and failures for a given group
 - ◇ 8.3.5 All login failures in the last few hours
 - ◆ 8.4 Other reports
 - ◇ 8.4.1 Idle Users for the last 180 days but created over 30 days
 - ◇ 8.4.2 List inactive users, locked, disabled, deleted and show creation date, login and last PIN change date
 - ◇ 8.4.3 List Users IDs, email, creation date, last login, last pin change and repository name
 - ◆ 8.5 Writing Your Own Reports
 - ◇ 8.5.1 Database Schema
 - ◇ 8.5.2 Report Definition Format
- 9 Troubleshooting

Overview

The Swivel Administration Console allows a limited number of reports to be generated and scheduled. This document outlines some of the information that can be queried through the reporting.

The **Audit Log** is where Swivel maintains an activity log for users. For more information on the Audit Log feature within Swivel, see the [Audit Log How To Guide](#).

For information on querying the Internal Swivel database see [Reporting Using Agent-XML How to Guide](#)

For information on querying a MySQL database see [MySQL Queries How To Guide](#) and for a MS SQL database see [MS SQL Queries How To Guide](#)

Also see [ReportingAPI](#)

Prerequisites

Swivel 3.8 - Reporting

Swivel 3.9 - Scheduled Reporting

Reporting access and restrictions

Swivel reports from a global database. Where access is restricted for Helpdesk users to specific repositories and groups it may be necessary to disable the reporting function for Helpdesk users. However reports can be produced by Administrative level access users creating scheduled reporting based on the required groups and making those reports available to the relevant Helpdesk users managing those groups such as through automated emails.

Producing Instant Reports

From the Swivel Administration console select Reporting then Instant.

Reporting

Select Report:

Report Description: No report selected

Rows per page

Depending on the report, additional parameters may be selected.

Reporting

Select Report:

Report Description: Lists users that have not logged in since a specified date

Rows per page

Report Parameters

Idle since [select date](#)

Click on Run Report to generate data.

Reporting

Select Report: 

Report Description: Lists users that have not logged in since a specified date

Rows per page

Report Parameters

Idle since [select date](#)

Report Result

1 rows found

#	Username	Last Login
1	test	

[Export as XML](#) [Export as CSV](#)

Clicking on *Export as XML* or *Export as CSV* allows the report to be saved from the browser.

Producing Scheduled Reports

From the Swivel Administration console select Reporting then Scheduled. Existing scheduled reports are listed. For information on creating custom schedules see [Schedule](#).

Scheduled Reports

Name	Report	Schedule	Format	Enabled		
inactive users	idleUsers	1 1 * * * ?	CSV	✓	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
never logged in	noconnect	0 2/15 * * * ?	XML	✓	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
list all users	allUsers	0 0/15 * * * ?	CSV	✓	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>

To create a new scheduled report click on *Add*

Scheduled Reports

Name	Report	Schedule	Format	Enabled		
New Report	failures		None	X	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
inactive users	idleUsers	1 1 * * * ?	CSV	✓	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
never logged in	noconnect	0 2/15 * * * ?	XML	✓	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>
list all users	allUsers	0 0/15 * * * ?	CSV	✓	<input type="button" value="Edit"/>	<input type="button" value="Delete"/>

Name: New Report

Report:

Schedule: Every at minutes past the hour

Format:

Filename:

Enabled

Enter the following information:

Name: Name for the report

Report: Select the required report

Schedule: How often the report is run

Format: The output format for the report, options are XML or CSV

Filename: The filename given to the report, for multiple reports to be run at the same time this should be unique, example *inactive_users_%d-%t.txt* creates a file called inactive_users_date-time.txt

Email Ticking this box enables the reporting by email. This function is available from Swivel 3.10 onwards. The email address is set under Policy/Reporting

Enabled Ticking this box enables the report to be run.

Apply the settings to save them.

Scheduled report file locations

Scheduled reports are stored within the Swivel installation, and varies depending upon the install type and version.

Swivel 3.9.1 onwards stores the reports in <swivel home>/swivel/reporting for Swivel appliances and software installations.

Swivel 3.9 stores the scheduled reports data in <path to pinsafe>/WEB-INF/reports

For an appliance <path to pinsafe> is: **/usr/local/tomcat/webapps/pinsafe**

For a Windows 3.9.1 installation, <swivel home> will depend on the user account under which Tomcat is running, and will typically be **C:\Users\Account**. You can check the actual directory from the PINsafe Administration Console Status page. Look for **Data Storage Root**.

For a Swivel 3.9 software install under Windows, <path to pinsafe> is usually: **C:\Program Files\Apache Software Installation\Tomcat 5.5\webapps\pinsafe**. Obviously, if you are using a different version of Tomcat, adjust accordingly.

Report date Format

The report date format can be set on the Swivel Administration console under Server/Language/Date Format

Available Reports

List all users

List all users with no activity since a given date

List all users with no activity within a given number of days

List login failures and self-resets since the last successful login

List creation time and latest login for all users

List users that have never logged in

List number of users hourly logged

Custom Reports

Always backup files before editing.

The report definition file is under the following locations:

Appliances after 3.9: /home/swivel/.swivel/conf/reports.xml

Appliances up to 3.9: /usr/local/tomcat/webapps/pinsafe/WEB-INF/conf/reports.xml

Software install after 3.9 <swivel home>/.swivel/conf/reports.xml

Software install up to 3.9 <path to Tomcat>/webapps/pinsafe/WEB-INF/conf/reports.xml

See above for details on finding the location of <swivel home>.

The default contents of this file at the time of writing is as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<reports>
  <report name="">
    <title>---- Please select a report to run ----</title>
    <description>No report selected</description>
  </report>
  <report name="allUsers">
    <title>List all users</title>
    <description>Lists all usernames in the PINsafe database</description>
    <headers>
      <header>Username</header>
    </headers>
    <fields>H</fields>
    <tables>PINSAFEJ</tables>
  </report>
  <report name="idleUsers">
    <title>List idle users</title>
    <description>Lists users that have not logged in since a specified date</description>
    <headers>
      <header>Username</header>
      <header>Last Login</header>
    </headers>
    <fields>U,H, A.D</fields>
    <tables>PINSAFEJ U LEFT OUTER JOIN PINSAFEN A ON U.G = A.A AND A.C = 0</tables>
    <query>A.D IS NULL OR A.D < ?</query>
    <params>
      <param name="idledate" type="Date" label="Idle since" />
    </params>
  </report>
  <report name="failures">
    <title>User fail count and reset count</title>
    <description>Lists the number of consecutive login failures and self-resets since the last successful login for all users</description>
    <headers>
      <header>Username</header>
      <header>Fail Count</header>
      <header>Reset Count</header>
    </headers>
    <fields>H, B, F</fields>
    <tables>PINSAFEJ</tables>
    <query />
  </report>
  <report name="connection">
    <title>Latest connection for all users</title>
    <description>Lists the creation time and last login time for all users</description>
    <headers>
      <header>Username</header>
      <header>Created</header>
      <header>Last Login</header>
    </headers>
    <fields>U,H, A1.D, A2.D</fields>
    <tables>PINSAFEJ U INNER JOIN PINSAFEN A1 ON U.G=A1.A AND A1.C=3 LEFT OUTER JOIN PINSAFEN A2 ON U.G=A2.A AND A2.C=0</tables>
    <query />
  </report>
  <report name="noconnect">
    <title>Users that have never logged in</title>
    <description>Lists all users that have never successfully logged into PINsafe</description>
    <headers><header>Username</header></headers>
    <fields>H</fields>
    <tables>PINSAFEJ</tables>
    <query>G NOT IN (SELECT DISTINCT A FROM PINSAFEN WHERE C=0)</query>
  </report>
</reports>
```

Adding new reports

Currently, requests for new reports can be made via your reseller to PINsafe support supportdesk@swivelsecure.com.

Once you've obtained a new report, the easiest way to add it to the configuration file above, is to use WinSCP. See the [WinSCP How To Guide](#) for further information on using WinSCP.

To add custom reports to the built-in reporting feature of PINsafe, you would append this to the reports.xml file. Insert new reports immediately before the last line:

```
</reports>
```

Once you have installed the report, go to Reporting -> Instant, and the new report should be available from the drop-down list.

You don't need to stop and restart Tomcat for the report to become available: the list is reloaded every time the report page is displayed.

We do not recommend writing your own reports, but for those who are competent with SQL and recognise that the reports are essentially XML-encoded fragments of SQL statements, you can get the full database schema for the Swivel database from [here](#).

NOTE: Any reports that reference the policy flags table, PINSAFEC, will not work with Sentry version 4.2 or later, and must reference the new status flags table, PINSAFES. Custom reports listed below that need changing are noted as such, and two alternatives are given.

Creation time, last login and number of logins

This report lists all successful logins within the last 30 days (the audit trail is only kept for 30 days by default, but can be set to a different value).

```
<report name="loginCount">
<title>Login count and latest login for all users</title>
<description>Lists the creation time, last login time and number of logins for all users</description>
<headers>
<header>Username</header>
<header>Created</header>
<header>Last Login</header>
<header>Login Count</header>
</headers>
<fields>U.H, MAX(A1.D), MAX(A2.D), COUNT(AU.G)</fields>
<tables>PINSAFEJ U INNER JOIN PINSAFEN A1 ON U.G=A1.A AND A1.C=3 LEFT OUTER
JOIN PINSAFEN A2 ON U.G=A2.A AND A2.C=0 INNER JOIN PINSAFEM AU ON U.G=AU.G AND AU.A=0
GROUP BY U.G, U.H</tables>
<query />
</report>
```

If you want to include users that have not logged in within the last 30 days, change "INNER JOIN PINSAFEM" to "LEFT OUTER JOIN PINSAFEM".

List all users, together with their login count

This report lists the following information:

First Name

Last Name

Username

Phone Number

Creation Date

Last Login Date

Login Count

Note: creation date if within length of audit time

This report has been tested on both Internal and MySQL databases.

```
<report name="user-login-count">
<title>User details with login count</title>
<description>Lists all users, together with their login count over the last 30 days, and last login time</description>
<headers>
<header>Username</header>
<header>First name</header>
<header>Last name</header>
<header>Phone</header>
<header>Created</header>
<header>Latest login</header>
<header>Login count</header>
</headers>
<fields>U.H, ATT1.C, ATT2.C, ATT3.C, ACT1.D, ACT2.D, AUD.CT</fields>
<tables><![CDATA[
PINSAFEJ U
LEFT OUTER JOIN PINSAFEP ATT1 ON U.G=ATT1.A AND ATT1.B='givenname'
LEFT OUTER JOIN PINSAFEP ATT2 ON U.G=ATT2.A AND ATT2.B='familyname'
LEFT OUTER JOIN PINSAFEP ATT3 ON U.G=ATT3.A AND ATT3.B='phone'
LEFT OUTER JOIN PINSAFEN ACT1 ON U.G=ACT1.A AND ACT1.C=3
LEFT OUTER JOIN PINSAFEN ACT2 ON U.G=ACT2.A AND ACT2.C=0
LEFT OUTER JOIN (SELECT G, COUNT(*) CT FROM PINSAFEM WHERE A=0 GROUP BY G) AUD ON U.G=AUD.G
]]></tables>
</report>
```

List users of a particular group showing their last login over the last month

```
<report name="connectbygroup">
  <title>Authentication by User in a Group</title>
  <description>List of users in a particular group, listing their logins over the last month</description>
  <headers>
    <header>Group</header>
    <header>Username</header>
    <header>Login Date</header>
  </headers>
  <fields>G.A, U.H, A.E</fields>
  <tables>PINSAFEM A INNER JOIN PINSAFEJ U ON A.G=U.G INNER JOIN PINSAFEI G ON U.G=G.B</tables>
  <query>G.A=? ORDER BY U.C</query>
  <params>
    <param name="group" type="String" label="Group Name" />
  </params>
</report>
```

You will be prompted for the group *name*; type in the actual name of the group, remember it's the Swivel group name, not the Active Directory FQDN.

List Inactive Users Within a Period By Group

Note that for it to work properly, set the Policy -> General, Audit Log length to a suitably large value (example 365 days)

```
<report name="idleUsersByGroup">
  <title>List Inactive Users Within a Period By Group</title>
  <description>List members of a given group who did not log in within a specified time</description>
  <headers>
    <header>Username</header>
    <header>Last Login</header>
  </headers>
  <fields>U.H, A.D</fields>
  <tables>PINSAFEJ U LEFT OUTER JOIN PINSAFEN A ON A.A = U.G AND A.C = 0 JOIN PINSAFEI G ON G.B = U.G AND G.A = ?</tables>
  <query>U.G NOT IN (SELECT G FROM PINSAFEM WHERE A=0 AND E &gt;= ? AND E &lt;= ?) </query>
  <params>
    <param name="group" type="String" label="Group" />
    <param name="startdate" type="Date" label="Start Date" />
    <param name="enddate" type="Date" label="End Date" />
  </params>
</report>
```

Total number of users

```
<report name="numberOfUsers">
  <title>Total number of users</title>
  <description>Report the total number of users in the database</description>
  <headers>
    <header>Num. Users</header>
  </headers>
  <tables>PINSAFEJ</tables>
  <fields>COUNT(*)</fields>
</report>
```

Total number of users in each group

```
<report name="groupMemberCount">
  <title>Count of users in each group by repository</title>
  <description>List the number of users in each group within each repository</description>
  <headers>
    <header>Repository</header>
    <header>Group</header>
    <header>Num. Users</header>
  </headers>
  <fields>R.B, G.A, COUNT(U.G)</fields>
  <tables>PINSAFEJ U JOIN PINSAFEL R ON U.I=R.A JOIN PINSAFEI G ON U.G=G.B GROUP BY R.B, G.A ORDER BY R.B, G.A</tables>
</report>
```

List users in each group

```
<report name="groupMembers">
  <title>List of users in each group by repository</title>
  <description>List the users in each group by repository</description>
  <headers>
    <header>Repository</header>
    <header>Group</header>
    <header>Users</header>
  </headers>
  <fields>R.B, G.A, U.C</fields>
  <tables>PINSAFEJ U JOIN PINSAFEL R ON U.I=R.A JOIN PINSAFEI G ON U.G=G.B ORDER BY R.B, G.A, U.C</tables>
</report>
```

List all OATH users

```
<report name="TokenUsers">
  <title>List of users allocated OATH tokens</title>
  <description>Lists all users that have been allocated OATH tokens</description>
  <headers>
    <header>User</header>
  </headers>
  <fields>U.H</fields>
  <tables>PINSAFEJ U JOIN PINSAFEQ T ON U.G=T.C</tables>
</report>
```

List users with Mobile permissions

```
<report name="MobileUsers">
  <title>List of users entitle to use mobile client</title>
  <description>Lists users that have the "Mobile" right</description>
  <headers><header>User</header></headers>
  <fields>U.H</fields>
  <tables>PINSAFEJ U JOIN PINSAFEB R ON U.G = R.B</tables>
  <query>R.A = 2</query>
</report>
```

List users with a provisioned Mobile device

```
<report name="AllocatedMobileUsers">
  <title>List of users with provisioned mobile clients</title>
  <description>Lists users that have provisioned their mobile clients</description>
  <headers><header>User</header></headers>
  <fields>U.H</fields>
  <tables>PINSAFEJ U JOIN PINSAFEO M ON U.G = M.C</tables>
</report>
```

List users with Mobile client rights that have not yet provisioned a device

```
<report name="UnprovisionedMobileUsers">
  <title>List of users that have not yet provisioned a mobile device</title>
  <description>Lists users that have the right to use a mobile client, but have not yet provisioned one</description>
  <headers><header>User</header></headers>
  <fields>U.H</fields>
  <tables>PINSAFEJ U JOIN PINSAFEB R ON U.G = R.B AND R.A=2</tables>
  <query>U.G NOT IN (SELECT C FROM PINSAFEO)</query>
</report>
```

List the number of login failures for all users

```
<report name="loginFailCount">
  <title>Count of failed logins for all users</title>
  <description>Lists the number of failed logins for each user, to the extent of the audit records</description>
  <headers>
    <header>Username</header>
    <header>Fail Count</header>
  </headers>
  <fields>I, Count(A)</fields>
  <tables>PINSAFEM</tables>
  <query>A=14 group by I order by I</query>
</report>
```

List of users that provisioned before a given date

```
<report name="oldProvisions">
  <title>List of users that provisioned before a given date</title>
  <description>Lists users that last provisioned their mobile app earlier than a specified date.</description>
  <headers>
    <header>Username</header>
    <header>Provision date</header>
  </headers>
  <params>
    <param name="cutoffdate" type="Date" label="Provisioned before"/>
  </params>
  <fields>U.H, A.D</fields>
  <tables>PINSAFEJ U JOIN PINSAFEO M ON U.G = M.C JOIN PINSAFEN A ON U.G = A.A and A.C=15</tables>
  <query>A.D &lt; ?</query>
</report>
```

List unprovisioned Mobile Users for a given date

```
<report name="Unprovisionedbydate" supporteddbs="mariadb,mysql,mssql">
  <title>List Unprovisioned Mobile Users older than given date</title>
  <description>Lists users, together with the latest date that they Provisioned</description>
  <headers>
    <header>Username</header>
    <header>Last Changed</header>
  </headers>
  <fields>U.H, U.C name, MAX(A.D) lastdate</fields>
  <tables>PINSAFEJ U JOIN PINSAFEB R ON U.G = R.B AND R.A=2; PINSAFEJ U JOIN PINSAFEN A on U.G=A.A and A.C in (1,2,3,6) group by A.A ha
  <query>U.G NOT IN (SELECT C FROM PINSAFEO)</query>
  <params>
    <param name="changedate" type="Date" label="Latest date" />
  </params>
</report>
```

List the number of login failures since a given date, for all users

```
<report name="loginFailCountSinceDate">
  <title>Count of failed logins since a given date for all users</title>
  <description>Lists the number of failed logins for each user, since a given date</description>
  <headers>
    <header>Username</header>
    <header>Fail Count</header>
  </headers>
  <fields>I, Count(A)</fields>
  <tables>PINSAFEM</tables>
  <query>A=14 and E=? group by I order by I</query>
  <params>
    <param name="sinceDate" type="Date" label="Cutoff Date" />
  </params>
</report>
```

List the last PIN change for all users

NOTE: this report has two different versions, as the Internal database doesn't support the features used by the first version. You may wish to include only the version that matches your database.

```
<report name="lastchanged" supporteddbs="mysql,mariadb,mssql">
  <title>List of PIN changes by date</title>
  <description>Lists all users, together with the latest date that their PIN was changed, or the creation date if never changed, starti
  <headers>
    <header>Username</header>
    <header>Last Changed</header>
  </headers>
  <fields>U.C name, MAX(A.D) lastdate</fields>
  <tables>PINSAFEJ U JOIN PINSAFEN A on U.G=A.A and A.C in (1,2,3,6) group by A.A order by lastdate</tables>
</report>

<report name="lastchanged" supporteddbs="internal">
  <title>List of PIN changes by date</title>
  <description>Lists all users, together with the latest date that their PIN was changed, or the creation date if never changed, starti
```



```

    <headers>
      <header>Username</header>
      <header>Last Changed</header>
    </headers>
    <fields>U.C name, Q.DD lastdate</fields>
    <tables>PINSAFEJ U JOIN (SELECT A, Max(D) DD FROM PINSAFEN where C in (1,2,3,6) group by A) Q on U.G = Q.A order by lastdate</tables>
  </report>

```

List PIN changes older than a given date

NOTE: as above, there are different versions for different databases.

```

<report name="lastchanged2" supporteddbs="mariadb,mysql,mssql">
  <title>List of PIN changes older than given date</title>
  <description>Lists users, together with the latest date that their PIN was changed, or the creation date if never changed, starting w
  <headers>
    <header>Username</header>
    <header>Last Changed</header>
  </headers>
  <fields>U.C name, MAX(A.D) lastdate</fields>
  <tables>PINSAFEJ U JOIN PINSAFEN A on U.G=A.A and A.C in (1,2,3,6) group by A.A having date &lt; ? order by lastdate</tables>
  <params>
    <param name="changedate" type="Date" label="Latest date" />
  </params>
</report>

```

```

<report name="lastchanged2" supporteddbs="internal">
  <title>List of PIN changes older than given date</title>
  <description>Lists users, together with the latest date that their PIN was changed, or the creation date if never changed, starting w
  <headers>
    <header>Username</header>
    <header>Last Changed</header>
  </headers>
  <fields>U.C name, Q.DD lastdate</fields>
  <tables>PINSAFEJ U JOIN (SELECT A, Max(D) DD FROM PINSAFEN where C in (1,2,3,6) group by A) Q on U.G = Q.A</tables>
  <query>Q.DD &lt; ? order by lastdate</query>
  <params>
    <param name="changedate" type="Date" label="Latest date" />
  </params>
</report>

```

Lists the number of logins for each hour

NOTE: For mariadb and mysql only

```

<report name="loginsByHour">
  <title>List of logins by hour</title>
  <description>Lists the number of logins for each hour</description>
  <headers>
    <header>Date/Time</header>
    <header>Count</header>
  </headers>
  <fields>HOUR(E), COUNT(*)</fields>
  <tables>PINSAFEM</tables>
  <query>A=0 GROUP BY HOUR(E)</query>
</report>

```

List number of logins for each day and hour for the last 7 days

```

<report name="loginsByDayHour">
  <title>List of logins by Day and Hour</title>
  <description>Lists the number of logins for each day and hour for the last 7 days</description>
  <headers>
    <header>Date</header>
    <header>Hour</header>
    <header>Count</header>
  </headers>
  <fields>DATE(E), HOUR(E), COUNT(*)</fields>
  <tables>PINSAFEM</tables>
  <query>A=0 AND E<CURDATE() AND E>DATE_SUB(CURDATE(), INTERVAL 7 DAY) GROUP BY DATE(E), HOUR(E)</query>
</report>

```

Enhancement to the previous report to include login failures and restrict to a single group

```

<report name="loginsByDayHourPerGroup">
  <title>List of logins by Day and Hour for a group</title>
  <description>Lists the number of logins or failures for each day and hour for the last 7 days</description>
  <headers>
    <header>Date</header>
    <header>Hour</header>
    <header>Count</header>
  </headers>
  <fields>DATE(E), HOUR(E), COUNT(*)</fields>
  <tables>PINSAFEM</tables>
  <query>(A=0 OR A=14) AND E<CURDATE() AND E>DATE_SUB(CURDATE(), INTERVAL 7 DAY) AND G IN (SELECT B FROM PINSAFEI WHERE A=?) GROU
  <params>
    <param name="group" type="String" label="Group" />
  </params>
</report>

```

List users that have been recently deleted

```

<report name="deletedUsers">
  <title>List of users that have been deleted recently</title>
  <description>Lists usernames that have been permanently deleted, but who still appear in the audit table</description>
  <headers><header>Username</header></headers>
  <fields>DISTINCT(I)</fields>
  <tables>PINSAFEM</tables>
  <query>I NOT IN (SELECT C FROM PINSAFEJ)</query>
</report>

```

Internal database reports

Edit reports.xml and search for the following line:

```
<report name="idleUsers2" supporteddbs="mysql,mssql,oracle">
```

If this line cannot be found, or there is no *supporteddbs* attribute, then it may be an older version, that does not support this.

Modify the above line as follows to add internal to the list of supported databases:

```
<report name="idleUsers2" supporteddbs="mysql,mssql,oracle,internal">
```

Internal database, lists users that have not logged in since a specified date

Add a query as follows, Insert this after the other <query ... lines and before <params>.

```
<query db="internal">A.D IS NULL OR {fn TIMESTAMPDIFF(SQL_TSI_DAY, CURRENT_DATE, A.D)} > ? </query>
```

Example

```
<query>A.D IS NULL OR A.D < ?</query>
<query db="internal">A.D IS NULL OR {fn TIMESTAMPDIFF(SQL_TSI_DAY, CURRENT_DATE, A.D)} > ? </query>
<params>
  <param name="idledate" type="Date" label="Idle since" />
</params>
```

Custom reports for multiple database types

The following reports will work with differing database types as the database is specified within the report as db="mysql db="mssql" db="oracle" db="internal"

Users marked as deleted

This report needs to be modified to work with Sentry version 4.2 onwards. Two different versions are given below:

Version 4.2 onwards:

```
<report name="deleted-users">
  <title>List of users marked as deleted</title>
  <description>List all users who are currently marked as deleted.</description>
  <headers><header>Username</header></headers>
  <fields>U.H</fields>
  <tables>PINSAFEJ U INNER JOIN PINSAFES F ON U.G=F.A</tables>
  <query>F.D &#amp; 1 = 1</query>
</report>
<report name="deleted-users-by-date">
  <title>List of users marked as deleted recently</title>
  <description>List all users who have been marked as deleted in the last #days.</description>
  <headers>
    <header>Username</header>
    <header>Date</header>
  </headers>
  <fields>U.H, A.D</fields>
  <tables>PINSAFEJ U INNER JOIN PINSAFEC F ON U.G=F.C AND F.B=4 AND F.D=1 INNER JOIN PINSAFEN A ON U.G=A.A AND A.C=10</tables>
  <query db="mysql">F.D &#amp; 1 = 1 AND A.D > DATE_SUB(NOW(), INTERVAL ? DAY)</query>
  <query db="mssql">F.D &#amp; 1 = 1 AND A.D > DATEADD(day, -, GETDATE())</query>
  <query db="oracle">F.D &#amp; 1 = 1 AND A.D > ADD_DAYS(SYSDATE, -?)</query>
  <query db="internal">F.D &#amp; 1 = 1 AND {fn TIMESTAMPDIFF(SQL_TSI_DAY, A.D, CURRENT_DATE)} &lt; ? </query>
  <params>
    <param name="days" type="Integer" label="Cutoff days" />
  </params>
</report>
```

Up to version 4.1.3:

```
<report name="deleted-users">
  <title>List of users marked as deleted</title>
  <description>List all users who are currently marked as deleted.</description>
  <headers><header>Username</header></headers>
  <fields>U.H</fields>
  <tables>PINSAFEJ U INNER JOIN PINSAFEC F ON U.G=F.C AND F.B=4 AND F.D=1</tables>
</report>
<report name="deleted-users-by-date">
  <title>List of users marked as deleted recently</title>
  <description>List all users who have been marked as deleted in the last #days.</description>
  <headers>
    <header>Username</header>
    <header>Date</header>
  </headers>
  <fields>U.H, A.D</fields>
  <tables>PINSAFEJ U INNER JOIN PINSAFEC F ON U.G=F.C AND F.B=4 AND F.D=1 INNER JOIN PINSAFEN A ON U.G=A.A AND A.C=10</tables>
  <query db="mysql">A.D > DATE_SUB(NOW(), INTERVAL ? DAY)</query>
  <query db="mssql">A.D > DATEADD(day, -, GETDATE())</query>
  <query db="oracle">A.D > ADD_DAYS(SYSDATE, -?)</query>
  <query db="internal">{fn TIMESTAMPDIFF(SQL_TSI_DAY, A.D, CURRENT_DATE)} &lt; ? </query>
  <params>
    <param name="days" type="Integer" label="Cutoff days" />
  </params>
</report>
```

Users whose PIN never expires

Version 4.2 onwards:

```
<report name="pin-never-expires">
  <title>List of users whose PIN never expires</title>
```

```

<description>Lists the names of users that have the "PIN never expires" flag set.</description>
<headers><header>Username</header></headers>
<fields>U.H</fields>
<tables>PINSAFEJ U INNER JOIN PINSAFES F ON U.G=F.A</tables>
<query>F.B=1</query>
</report>

```

The above report can be modified to show users who are marked as disabled, locked, deleted, inactive or require a PIN change, by changing the query as follows:

- User must change PIN after next login: `<query>F.C=1</query>`
- User disabled: `<query>F.D & 2 = 2</query>`
- User locked (any reason): `<query>F.D & 124 < 0</query>`
- User is marked as deleted: `<query>F.D & 1 = 1</query>`

Up to version 4.1.3

```

<report name="pin-never-expires">
<title>List of users whose PIN never expires</title>
<description>Lists the names of users that have the "PIN never expires" flag set.</description>
<headers><header>Username</header></headers>
<fields>U.H</fields>
<tables>PINSAFEJ U INNER JOIN PINSAFEC F ON U.G=F.C</tables>
<query>F.B=3 AND F.D=1</query>
</report>

```

The above report can be modified to show users who are marked as disabled, locked, deleted, inactive or require a PIN change, by changing the value of F.B in the query as follows:

- 0 - User disabled
- 1 - User locked
- 2 - User must change PIN after next login
- 3 - User's PIN never expires
- 4 - User is marked as deleted
- 5 - User is inactive

All User Logins, including login location

NOTE: this will only report logins as far back as audit records are retained (by default, 30 days). Also, the location field is not always reported. Most AgentXML integrations (e.g. OWA) do not report the user's location. RADIUS-based integrations will report the calling workstation ID if the NAS reports it, or else the NAS IP address.

```

<report name="connections">
<title>All recent connections for all users</title>
<description>Lists the login times and location (where available) for all users, within the audit retention time</description>
<headers>
<header>Username</header>
<header>Created</header>
<header>Last Login</header>
<header>Location</header>
</headers>
<fields>U.H, A1.D, AU.E, AU.B</fields>
<tables>PINSAFEJ U INNER JOIN PINSAFEN A1 ON U.G=A1.A AND A1.C=3 LEFT OUTER JOIN PINSAFEM AU ON U.G=AU.G AND AU.C=0</tables>
<query>1 ORDER BY U.H, AU.E DESC</query>
</report>

```

All user logins and failures for a given group

This reports user logins and failures for a given period from the current date and a given group.

NOTE: this will only report logins as far back as audit records are retained (by default, 30 days). Also, the location field is not always reported. Most AgentXML integrations (e.g. OWA) do not report the user's location. RADIUS-based integrations will report the calling workstation ID if the NAS reports it, or else the NAS IP address. Details also may not have been populated.

```

<report name="LoginDetails">
<title>Details of User logins</title>
<description>List of all logins and login failures within a given time for a given group</description>
<headers>
<header>Username</header>
<header>Activity</header>
<header>Date/Time</header>
<header>Location</header>
<header>Details</header>
</headers>
<fields>I, if (A=0, 'login', 'failed'), E, B, C</fields>
<tables>PINSAFEM</tables>
<query db="mariadb,mysql">(A=0 OR A=14) AND DATEDIFF(CURRENT_DATE(), E) < ? AND G IN (SELECT B FROM pinsafei WHERE A=?) ORDER BY I, E
<params>
<param name="days" label="Number of days" type="Integer" />
<param name="group" label="Group" type="String" />
</params>
</report>

```

All login failures in the last few hours

This report gives a summary of the number of login failures for each user within a given time from the current time. The time difference is given in hours.

This report will only go back as far as audit records are retained (by default 30 days), so specifying a very large number of hours will not give complete results.

```

<report name="RecentLoginFailCount">
<title>Count of failed logins in the last x hours</title>
<description>Lists the number of failed logins for each user in the last few hours</description>
<headers>
<header>Username</header>
<header>Fail Count</header>
</headers>
<fields>I, Count (A)</fields>
<tables>PINSAFEM</tables>
<query db="mariadb,mysql">A=14 and E>DATE_SUB(CURRENT_TIMESTAMP(), INTERVAL ? HOUR) group by I order by I</query>

```

```

<params>
  <param name="cutoffHours" type="Integer" label="# hours"/>
</params>
</report>

```

Other reports

The following reports were tested and made available regarding specific needs:

Idle Users for the last 180 days but created over 30 days

```

<report name="idleusers180createdover30">
  <title>Users who never logged in over 180 days but created over 30</title>
  <description>List the number of users who have never logged in over the last 180 days but created more than 30 days ago</description>
  <headers>
    <header>Username</header>
  </headers>
  <fields>U.H</fields>
  <tables>PINSFAFEJ U JOIN PINSAFEN A1 ON U.G = A1.A AND A1.C = 3 LEFT OUTER JOIN PINSAFEN A2 ON U.G = A2.A AND A2.C=0</tables>
  <query><![CDATA[(A1.D &lt; DATE_SUB(NOW(), INTERVAL 30 DAY)) AND (A2.D IS NULL OR A2.D &lt; DATE_SUB(NOW(), INTERVAL 180 DAY))]]></query>
</report>

```

List inactive users, locked, disabled, deleted and show creation date, login and last PIN change date

Version 4.2 onwards:

```

<report name="inactivelockeddisabledUsers">
  <title>Users inactive, locked, disabled, deleted and show creation, login, PIN change date</title>
  <description>List all Users inactive, locked, disabled, deleted and show creation, login, PIN change date</description>
  <headers>
    <header>UID</header>
    <header>REPOS</header>
    <header>DISABLED</header>
    <header>LOCKED</header>
    <header>DELETED</header>
    <header>INACTIVE</header>
    <header>CREATED</header>
    <header>LOGGEDIN</header>
    <header>PINCHANGE</header>
  </headers>
  <fields><![CDATA[U.H, R.B, S.D & 2 = 2, S.D & 116 <> 0, S.D & 1 = 1, S.D & 8 = 8, A1.D, A2.D, A3.D]]></fields>
  <tables><![CDATA[PINSFAFEJ U JOIN PINSAFEL R ON U.I=R.A JOIN PINSAFES S ON U.G=S.A LEFT OUTER JOIN PINSAFEN A1 ON U.G=A1.A AND A1.C=3 L]]></tables>
</report>

```

Up to version 4.1.3:

```

<report name="inactivelockeddisabledUsers">
  <title>Users inactive, locked, disabled, deleted and show creation, login, PIN change date</title>
  <description>List all Users inactive, locked, disabled, deleted and show creation, login, PIN change date</description>
  <headers>
    <header>UID</header>
    <header>REPOS</header>
    <header>DISABLED</header>
    <header>LOCKED</header>
    <header>DELETED</header>
    <header>INACTIVE</header>
    <header>CREATED</header>
    <header>LOGGEDIN</header>
    <header>PINCHANGE</header>
  </headers>
  <fields>U.H, R.B, S1.D, S2.D, S3.D, S4.D, A1.D, A2.D, A3.D</fields>
  <tables><![CDATA[PINSFAFEJ U JOIN PINSAFEL R ON U.I=R.A LEFT OUTER JOIN PINSAFEC S1 ON U.G=S1.C AND S1.B=0 LEFT OUTER JOIN PINSAFEC S]]></tables>
</report>

```

List Users IDs, email, creation date, last login, last pin change and repository name

```

<report name="allUsersdataLastLoginandRepo">
  <title>Users IDs, email, date, last login, last pin change and repo name</title>
  <description>Lists all the Users ids, email address, created date, last login, last pin change and repository name</description>
  <headers>
    <header>Username</header>
    <header>RepoName</header>
    <header>Email</header>
    <header>CreateDate</header>
    <header>LoginDate</header>
    <header>PINChange</header>
  </headers>
  <fields>U.H, U.E, T.A, A1.D, A2.D, A3.D</fields>
  <tables><![CDATA[PINSFAFEJ U LEFT OUTER JOIN PINSAFEP T ON U.G = T.A AND T.B='email' LEFT OUTER JOIN PINSAFEN A1 ON U.G = A1.A AND A1.C=3 L]]></tables>
</report>

```

Writing Your Own Reports

This section is intended only for customers who are familiar with writing SQL queries, and with XML.

Database Schema

While writing your report, refer to the [Database Schema](#) page for reference on the database table and column names used by Swivel.

Report Definition Format

A report definition is an xml element, which includes elements of a SQL query, plus display information. The format is:

```

<report name="reportName">
  <title>Report Title</title>
  <description>Report description</description>
  <headers>
    <header>header1</header>
    ...

```

```
</headers>
<fields>SQL-fields</fields>
<tables>SQL-tables-and-joins</tables>
<query>SQL-query</query>
<params>
  <param name="param-name" label="param-label" type="String|Integer|Date" />
  ...
</params>
</report>
```

The report name is used internally, but must be unique.

<title>, <description> and <headers> are display elements: the title is displayed in the drop-down and at the top of the report, while the description is shown when the report is selected. Headers are individual column headers, and the number of headers should match the number of columns in the field list.

The SQL query is built as

```
SELECT SQL-fields FROM SQL-tables-and-joins WHERE SQL-query
```

The <query> element is optional: if missing, no WHERE clause is added. There are no elements for ORDER BY or GROUP BY, but you can add "ORDER BY" or "GROUP BY" to the query element, or to the tables element if there is no query.

The <params> elements are used to set values for replaceable parameters within the query. <param> elements are applied in the order listed. The name attribute is used to identify the parameter internally. The label attribute is displayed on-screen when requesting parameter values. The type attribute must be "String", "Integer" or "Date" - these values are case-sensitive.

Troubleshooting

Versions 3.8 to 3.9.3 may produce a HTTP Status 500 error when selecting a new report:

java.lang.IllegalArgumentException: fromIndex(50) > toIndex(0)

java.util.SubList.<init>(Unknown Source)

java.util.AbstractList.subList (Unknown Source)

com.swiveltechnologies.pinsafe.server.reporting.ReportResult.getRows (ReportResult.java:122)

This is caused by last page you were on when you used a report. For example if you were on page 4 of one particular report, but then attempt to access another report which does not contain 4 pages, it will attempt to access page 4 of that report instead of page 1 and fail with the error you sent.

So the solution is to upgrade, or as a workaround, make sure that you request page 1 of a report before leaving the reporting page. Closing down the web browser should also alleviate this.