SQL as a data source How To Guide

Contents

SQL as a data source How To Guide

Requirements

- PINsafe 3.7 or later.
- Relevant JDBC driver for SQL Server
- SQL Server Account with full read permissions on the database containing the users. (PINsafe does not support using Windows authentication).

NOTE: This article references Microsoft SQL Server and Oracle 10g, it is possible to use other database servers.

Microsoft SQL Driver

Microsoft US English JDBC driver for JRE 6 sqljdbc4.jar: [[1]]

Microsoft US English JDBC driver for JRE 5 sqljdbc.jar: [[2]]

Click on the Download button to show the EULA. At the bottom of this, it gives you the option to download the Windows or Linux version. The only difference is the format of the download: the actual application is the same for both, so download the Windows version, which gives you a self-extracting executable. Extract this to a temporary folder, since we only need one file from the entire bundle. Go to this folder, and into the one sub-folder (named enu on the US English version). Copy the sqljdbc.jar or sqljdbc4.jar to the PINsafe server, into the PINsafe webapp folder under WEB-INF/lib. If this is an appliance, you can upload using Webmin. Make sure that you set both the file owner and group are set to "swivel".

Oracle SQL Driver

Download the appropriate Oracle driver i.e. ojdbc6.jar from http://www.oracle.com/technetwork/database/features/jdbc/index-091264.html

Copy the file to the PINsafe server, into the PINsafe webapp folder under WEB-INF/lib. If this is an appliance, you can upload using Webmin. Make sure that you set both the file owner and group are set to "swivel".

Creating the Data Source

1. Check under Repository -> Types whether there is a type called "Database". If you have upgraded from an earlier version of PINsafe, it may not be there, in which case you will need to add it. Create a new type by entering the following data in the empty fields at the bottom of the list: Identifier = Database, Class = com.swiveltechnologies.pinsafe.server.user.repository.DBRepository. Click Apply.

2. Under Repository -> Servers, create a new repository. Enter the Repository Name, then under Repository Type you will see the new type "Database". Select that, then click Apply.

3. The next job is to configure the new repository.

- JDBC Driver: The JDBC driver class. Enter com.microsoft.sqlserver.jdbc.SQLServerDriver. For other database servers, check your documentation.
- Database URL: The URL of the database as needed by the JDBC driver. This needs to be in the form:

jdbc:sqlserver://<server>:1433;databaseName=<database>

or

jdbc:oracle:thin:@<server>:1523/<inst>

For other database servers, check your documentation.

- Database login user: The username you have created for accessing the database.
- Database login password: The password of the user account.
- User details table: The name of the table containing the users.
- User ID field: The name of the field containing the user ID.
- Username field: The name of the field containing the username.
- Initial PIN field: (optional) The name of the field containing the initial PIN.
- Initial password field: (optional) The name of the field containing the initial password.
- Disabled flag field: (optional) The name of the field containing the disabled flag.
- Group membership table: The name of the group membership table.
- Membership userid field: The name of the userid field in the membership table.
- Membership group field: The name of the group field in the membership table.
- Import disabled state: Whether or not to import users' disabled status.
- Synchronisation schedule: The user sync schedule. It is suggested that you leave this as "NEVER" to start with, and use manual sync until you are happy that the user import is working correctly.

You can get more information on the format of the database URL from the documentation included with the driver, but generally you just need to replace <server> and <database> with the SQL Server name and database name. You may need to change the port (1433) if the server is not the default instance of SQL Server on the machine. You may also find that you need to enable TCP/IP connections on the SQL Server, and also the SQL Server Browser service.

With regard to the database tables, it is assumed that you have a table of users, containing the following fields:

- userid : the user primary key
- username : the name that PINsafe needs to import. This can be the same field as userid.

The remaining fields are optional

- Email : the user's email address.
- Phone : the user's mobile phone number.
- PIN : the initial PIN. This is only imported for new users.
- Password : the initial PINsafe password, as for PIN.
- Disabled flag : set to 1 if the user should be disabled. Set import disabled to use this.

The actual names of the fields are entered as described above, except see below for email and phone.

Secondly, you will need to have a group membership table. If you don't have one of these, for example if all users should be imported with the same rights, then you will need to create it and populate it as follows:

- Userid : the ID of the user
- Group : the name of the group the user is a member of.

If your users table has a field which is suitable to be used as the group identifier, then you can use the same table for users and group membership. In this case, the group userid field will be the same as the user userid field.

4. Under Repository -> Attributes, set the email and phone attributes as the names of the appropriate database fields. You can also configure any other user attributes you want to import.

5. Under Repository -> Groups, set the group value for any groups you are using to the value of the group field from the membership table.

As an example, if you want to import all users as members of the PINsafeUsers group, and you do not yet have a membership table, then simply put "pinsafe" in the group definition for PINsafeUsers for the database repository. Then create a new table called membership, with two fields, userid and group. userid should have the same type as the userid field from the users table, and group should be varchar. Finally, populate the table with the following statement:

```
INSERT INTO membership (userid, group) SELECT userid, 'pinsafe' FROM users
```

6. Now go to User Administration, select the database repository, and click User Sync. The users should be imported correctly.

WARNING: PINsafe makes no attempt to check whether column names are reserved words, or to escape them if they are. You need to ensure that the column name is escaped if necessary. The method for doing this will depend on the database server you are using. For example, if the column name is "group", in SQL Server you must enter it as "[group]" (minus the quotation marks).

Sample Oracle Integration

Repository <repository name> Settings

JDBC Driver: oracle.jdbc.driver.OracleDriver

Database login user: jdbc:oracle:thin:@<server>:1523/SIP

Oracle service account Username: oracpinsafe

Oracle password: ********

Oracle SQL port: 1523

User details table: FND_USER

User ID field: USER_ID

Username field: USER_NAME

Group membership table: membership

Membership userid field: USER_ID

Membership group field: GROUP_ID

Repository/Group Settings

PINsafe Users pathname: pinsafe

Transport/Attribute Settings

Email Transport Attribute: EMAIL_ADDRESS

Further Information

Please contact Swivel Secure support for further information.