Sans Pareil
TECHNOLOGIES

UNIX Password Application
Application Manual
Version 1.0

Prepared by
Rakesh Vidyadharan

Prepared for

Dated: April 24, 2007

Contents

List of Tables

1 Introduction

A simple application using Jakarta Commons Net that can be used by UNIX system users to change their password. This was developed as an ancillary to SPT Web Mail Application to provide the means for Sans Pareil Technologies, Inc. employees to manage their system password.

2 Features

The SPT Password Management Application application behaves in a similar fashion to most password management applications. The following steps describe the process by which a user can modify their password.

1. Logon to the system using their `username` and existing `password`.
2. Change their password by entering the following information:
   - 2.1. Enter their `current password`.
   - 2.2. Enter their desired `new password` twice.
   - 2.3. Use the Suggest Password button to let the system generate a randomly generated secure password.
   - 2.4. Checks the quality of the new password using the following rules:
     - 2.4.1. Password is at least 6 characters in length.
     - 2.4.2. Password contains at least 2 letters.
     - 2.4.3. Password contains at least 1 non-alphabetic character.
     - 2.4.4. Password does not contain all the characters in `username` in any sequence.
     - 2.4.5. Password does not contain all the characters in `current password` in any sequence.

3 Installation

The SPT Password Management Application is easy to install. Table 1 shows the versions of Java/J2EE that are necessary for the system. Table 2 shows the environment in which the system has been tested.

Table 1: Java Requirements

Software	Version	Description
Java	1.5	Minimum version of the JRE that is required.
Servlet	1.3	Minimum version of the Servlet API that is required.

The following steps describe the process of deploying the SPT Password Management Application to a J2EE/Servlet container. All files referred to in the instructions are available in the distribution.

1. Enable `telnet` server on the server on which the user accounts are maintained.

2. Copy the `config/password.xml` file to a convenient location on the server. For example. Sans Pareil Technologies, Inc. stores the file at `/var/data/system/password.xml`. Edit the file to match your environment (see section 3.1 for details).

3. Build the application. The application will be created as `deploy/password.war`.

   3.1. Edit `ant.properties` to match your system.

   3.2. Execute the `ant` build script.

   3.3. Run `ant test` to run the unit tests. This can also be used to verify connectivity and compatibility with your UNIX system.

4. Configure a JVM system property named `password.properties.file` for the container. For Tomcat for example, add a line similar to the following to `$CATALINA_BASE/bin/catalina.sh`

```
JAVA_OPTS="$JAVA_OPTS -Dpassword.properties.file=/var/data/system/password.xml"
```

5. Deploy the `password.war` built to the container. You may have to rename the war file to something else (for example, system.war) if you wish to customise the root path at which the application is to be accessed.

Table 2: Test Environment

Software	**Version**	**Description**
Tomcat	5.5.17	Servlet container used to deploy the application.
Solaris	10	The operating system used on Sans Pareil Technologies, Inc. server.
Mac OS X	10.4 (Tiger)	The client operating system used for testing.

3.1 Configuration File

The `telnet` server connection properties are configured in the `password.xml` file. Table 3 describe the properties configured in the file:

Table 3: Configuration Properties

Property	**Description**
`server`	The server to connect to via `telnet`.
`port`	The network port to connect to.
`login`	The login prompt presented by the telnet server.
`password`	The password prompt presented by the telnet server.
`prompt`	The shell prompt character to use to determine end of command transaction.

4 Customisation

The application may be customised to meet difference requirements to a fair degree.

4.1 Localisation

Most of the text displayed in the application[1] are configured for each UI Component in a properties file[2]. The values in the file may be modified to suit individual tastes. Copies of the file for other languages may be created as well which will be loaded based upon client browser language preference using regular Java property file loading rules.

4.2 Dimensions

The default dimensions used to build the UI components are specified for each UI Component in a properties file[3]. The default sizes, as well as maximum values allowed in TextComponent[4] are specified in this file. You can edit the values in this file to suit your preferences.

4.3 Password Rules

The rules used to enforce password quality are defined in the following classes. You can modify the classes to suit your requirements.

- `com.sptci.util.PasswordGenerator`
- `com.sptci.system.PasswordController`

5 Licences

The SPT Password Management Application is distributed under a Apache 2.0[5] licence. The third-party licences that apply to the system are included under the `config/licenses` directory in the distribution. Table 4 lists the licenses and a brief description of their use in the application.

6 FAQ's

Sections 6.x list some questions that may be frequently asked by people attempting to use the SPT Password Management Application.

[1] Labels, titles, . . . not including content of the message or folder.

[2] Located under `config/resource/localisation/Configuration.properties` in the distribution.

[3] Located under `config/resource/localisation/Dimensions.properties` in the distribution.

[4] http://www.nextapp.com/platform/echo2/echo/doc/api/public/app/nextapp/echo2/app/text/TextComponent.html

[5] http://www.apache.org/licenses/LICENSE-2.0

Table 4: Licence Files

File	Description
password.txt	This file contains the licence under which the SPT Password Management Application is distributed. The licence applies to all source code distributed by Sans Pareil Technologies, Inc. as part of the system.
echo2.txt	This file contains the licence under which the Echo2 library is distributed. Echo2 is the primary framework utilised to develop the User Interface.. This is the same licence under which the associated EchoPointNG, Echo2FileTransfer, and Echo2Consultas libraries are distributed. Please note that some source files in the EchoPointNG library were modified to meet the requirements of the system. The modified source files have been included as config/epng.tar file in the distribution.
commonsnet.txt	This contains the licence under which the Jakarta Commons Net library is distributed.

6.1 On what systems will the application work?

The application uses telnet to connect to the server. It uses the passwd UNIX command to change the password. Thus, the server on which the user accounts are maintained will need to be a UNIX system. The application itself can be deployed on any platform.

6.2 Why do you use telnet?

The only reason is that a TelnetClient is freely available through the Jakarta Commons Net library. I could not find a way to run passwd through java.lang.Runtime.exec method.

6.3 Isn't telnet insecure?

Yes, however since the appliation server will almost certainly have access to the telnet server over a local network the security vulnerabilities can be easily controlled. For maximum security deploy the application on the server used to maintain user accounts.