## Contents

1

### CHAPTER 1 Preparing for Installation  

Hardware and software requirements .......................................................... 2  
Installation overview ................................................................. 5  
Installation options for 4.4 ................................................................. 5  
Installation procedures ................................................................. 6  
Preparing the database ................................................................. 8  
  Preparing a PostgreSQL database .................................................. 8  
  Preparing an Oracle database ......................................................... 9  
Preparing the SourceForge server .................................................. 11  
  Preparing the SourceForge server on RedHat or SUSE Linux ................. 11  
Preparing the SCM servers .......................................................... 13  
  Installing third party packages ...................................................... 13  
  Preparing a SourceForge SCM Integration Server ............................. 13  
  Preparing a CVS server .............................................................. 14  
  Configuring LDAP for a CVS Integration Server ............................ 15  
Preparing a Subversion server ..................................................... 20  
Preparing a Perforce server ......................................................... 22  
Editing the SourceForge install configuration file ............................ 24

### CHAPTER 2 Installing and Configuring SourceForge  

Running the installation program ...................................................... 30  
Finishing the SourceForge installation ............................................. 32  
Configuring Apache ................................................................. 32
### CHAPTER 3 SourceForge Administration

- Finishing SCM installation ........................................... 33
- Configuring LDAP integration ....................................... 34
- Before you begin ...................................................... 34
- Configuring SourceForge for LDAP authentication .............. 35
- Adding additional SCM servers ..................................... 38
- Running the SCM configuration script ............................. 38
- Backups and restoration ................................................ 40
- JBoss ........................................................................... 40
- CVS ............................................................................. 41
- Subversion ..................................................................... 41
- PostgreSQL backups, restoration, and optimization ............ 41
- Oracle .......................................................................... 42
- Server logs ...................................................................... 43
- JBoss ............................................................................. 43
- SCM (CVS, Subversion, and Perforce) ............................... 43
- Email ............................................................................ 44
- Search ........................................................................... 44
- PostgreSQL .................................................................... 45
- Oracle ............................................................................ 45
- SourceForge security ..................................................... 47
- J2EE Architecture .......................................................... 47
- Key elements of security ................................................. 48
- Protection of sensitive data ............................................. 48
- Effective data access control ......................................... 49
- Activity tracking ............................................................. 50
- CERT advisories ............................................................. 50
- Summary ........................................................................ 52
- Firewalling, operating system, and Apache security .......... 53
- Generating Apache SSL certificates ................................. 56
- Further securing HTTPS .................................................. 57
- Preventing HTTP Trace/Put/Delete methods ..................... 58
- Recommended firewall configuration .............................. 59
- Password security .......................................................... 59
- Third party application support limitations ..................... 60
- CVS .............................................................................. 60
- Discussion forum threading .......................................... 60
- Support for other third party applications ...................... 61
CHAPTER 1

Preparing for Installation

This guide provides instructions for installing SourceForge® Enterprise Edition 4.4. It also includes instructions for preparing and integrating one or more CVS, Subversion, or Perforce SCM servers with your SourceForge installation.

SourceForge Enterprise Edition 4.4 runs on the JBoss application server and the following database servers:

- PostgreSQL
- Oracle

This chapter contains the following information:

- “Hardware and software requirements” on page 2
- “Installation overview” on page 5
- “Preparing the database” on page 8
- “Preparing the SourceForge server” on page 11
- “Preparing the SCM servers” on page 13
- “Editing the SourceForge install configuration file” on page 24
Hardware and software requirements

Hardware requirements

The following minimum hardware is recommended for the server on which SourceForge Enterprise Edition 4.4 is installed, and the server on which the database is installed.

It is highly recommended that SourceForge and the database reside on separate physical servers.

- 2 x CPU 2GHz
- 2 GB RAM
- 20GB hard drive

Note - Required hard drive capacity depends directly upon the estimated amount of document and file release uploads.

Software requirements

The following table lists the configurations on which SourceForge Enterprise Edition 4.4 has been fully tested.

<table>
<thead>
<tr>
<th>Database</th>
<th>Application Server</th>
<th>Application Server OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>PostgreSQL 8.2</td>
<td>JBoss 3.2.6</td>
<td>• RedHat Enterprise Server 3 U8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• RedHat Enterprise Linux 4 U3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• Novell SUSE Linux Enterprise Server 9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• CentOS4.3</td>
</tr>
<tr>
<td>Oracle 9.2.0.6¹</td>
<td>JBoss 3.2.6</td>
<td>• RedHat Enterprise Server 3 U8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• RedHat Enterprise Linux 4 U3</td>
</tr>
</tbody>
</table>

¹Requires the Oracle 10g client on the application server.
Supported Browsers
The following browsers have been fully tested with SourceForge Enterprise Edition 4.4:

- Mozilla Firefox 1.5.x
- Mozilla Firefox 2.0.0.4
- Internet Explorer 6.0
- Internet Explorer 7.0

Supported SCM Tools
The following Software Configuration Management (SCM) tools have been fully tested with SourceForge Enterprise Edition 4.4:

- CVS 1.11.17
- Subversion 1.2.3 (BDB & FSFS) on RedHat Enterprise Server 3
- Subversion 1.3.1 (FSFS) on SUSE Linux Enterprise Server 9
- Subversion 1.4.3 (FSFS) on RedHat Enterprise Linux 4
- Subversion 1.4.3 (FSFS) on CentOS4
- Perforce 2006.2

Supported Java SDK
The following Java SDK has been fully tested with SourceForge Enterprise Edition 4.4:

- Java SDK 1.5.0_12

Supported Oracle client
The following Oracle client has been fully tested with SourceForge Enterprise Edition 4.4:

- Oracle 10gR2 - 10.2.0.1.0 Standard Edition (the Express Edition is not supported)

Supported Microsoft® Applications
The following Microsoft applications have been fully tested with SourceForge Enterprise Edition 4.4:

- Microsoft® Project 2002 (with Service Pack 1) on WinXP\textsuperscript{1} and Win2000\textsuperscript{2}
- Microsoft® Project 2003 (with Service Pack 1) on WinXP\textsuperscript{1} and Win2000\textsuperscript{2}
- Microsoft® Office XP (with Service Pack 3) on WinXP\textsuperscript{1} and Win2000\textsuperscript{2}
- Microsoft® Office 2003 (with Service Pack 1) on WinXP\textsuperscript{1} and Win2000\textsuperscript{2}

\textsuperscript{1}With Service Pack 2
\textsuperscript{2}With Service Pack 4
Hardware and software requirements

**Supported LDAP Application**

The following LDAP (Lightweight Directory Access Protocol) application has been fully tested with SourceForge Enterprise Edition 4.4.

- OpenLDAP 2.0.27
Installation overview

A SourceForge Enterprise Edition 4.4 installation consists of the following components:

1. One main SourceForge server (the application) running on JBoss and either the Postgres or Oracle database, and
2. Any number of CVS, Subversion, or Perforce SCM servers.
   SCM installations can be installed on the SourceForge server, or on one or more different servers.

Installation options for 4.4

The SourceForge Enterprise Edition installer has been updated to allow you to install SourceForge and your SCM servers together or separately, on one or more physical servers.

You must define the components that the installer installs by configuring the installation variables in the following configuration file:

<SOURCEFORGE_INSTALLER_DIR>/install-saturn.conf

See “Editing the SourceForge install configuration file” on page 24 for details on configuring the installation variables.

You can install SourceForge and your SCM servers in one of the following four ways:

1. SourceForge and all SCM servers on the same physical server.
   Complete the configuration for SourceForge and all of the SCM installations first, then run the installer only once to install all components.

2. SourceForge on one physical server, and all SCM servers on a different physical server.
   Run the installer once to install SourceForge, then again to install one or more SCM servers on a different physical server.

3. New SCM server on standalone SourceForge physical server.
   To install one or more SCM servers on a physical server on which you have already installed SourceForge, but no SCM servers, run the installer again to install the SCM servers.

4. Additional SCM server.
   To install an additional SCM server on a physical server on which you have already installed one or more SCM servers, you must run the configureScm.sh script that prompts you for the necessary SCM configuration information and installs the SCM server. See “Running the SCM configuration script” on page 38.
Installation procedures

This section provides the installation steps for each of the installation options listed in “Installation options for 4.4” on page 5.

1. SourceForge and all SCM servers on the same physical server

To install SourceForge and one or more SCM servers on the same physical server, do the following:

- Prepare the database. See “Preparing the database” on page 8.
- Prepare the SourceForge server. See “Preparing the SourceForge server” on page 11.
- Prepare one or more SCM servers. See “Preparing the SCM servers” on page 13.
- Edit the configuration file to identify the components to be installed. See “Editing the SourceForge install configuration file” on page 24.
- Run the installation program. See “Running the installation program” on page 30.
- Finish the SourceForge installation. See “Finishing the SourceForge installation” on page 32.
- Configure Apache. See “/etc/httpd/conf/httpd.conf” on page 71.
- (Optional) Configure LDAP integration. See “Configuring LDAP integration” on page 34.

2. SourceForge on one physical server, and all SCM servers on a different physical server

To install SourceForge on one physical server, do the following:

- Prepare the database. See “Preparing the database” on page 8.
- Prepare the SourceForge server. See “Preparing the SourceForge server” on page 11.
- Edit the configuration file to identify that you are installing only SourceForge. See “Editing the SourceForge install configuration file” on page 24.
- Run the installation program. See “Running the installation program” on page 30.
- Finish the SourceForge installation. See “Finishing the SourceForge installation” on page 32.
- Configure Apache. See “/etc/httpd/conf/httpd.conf” on page 71.
- (Optional) Configure LDAP integration. See “Configuring LDAP integration” on page 34.

Then do the following to install one or more SCM servers on a different physical server:

- Prepare one or more SCM servers. See “Preparing the SCM servers” on page 13.
- Edit the configuration file to identify the SCM servers to be installed. See “Editing the SourceForge install configuration file” on page 24.
- Run the installation program. See “Running the installation program” on page 30.
- Finish the SCM installation. See “Finishing SCM installation” on page 33.
3. **New SCM server on standalone SourceForge physical server**
   To install one or more SCM servers on a physical server on which you have already installed SourceForge, but no SCM servers, do the following:
   - Prepare one or more SCM servers. See “Preparing the SCM servers” on page 13.
   - Edit the configuration file to identify the SCM servers to be installed. See “Editing the SourceForge install configuration file” on page 24.
   - Run the installation program. See “Running the installation program” on page 30.
   - Finish the SCM installation. See “Finishing SCM installation” on page 33.

4. **Additional SCM server**
   To install an additional SCM server on a physical server on which you have already installed one or more SCM servers, do the following:
   - Prepare one or more SCM servers. See “Preparing the SCM servers” on page 13.
   - Run the SCM installation script. See “Running the SCM configuration script” on page 38.
Preparing the database

SourceForge Enterprise Edition 4.4 requires one of the following databases:

- PostgreSQL 8.2
- Oracle 9.2.0.6 with Oracle 10g client (10.2.0.1.0)

This section provides instructions on preparing your database for SourceForge installation. It does not contain detailed database installation instructions. For detailed database installation instructions, see the user documentation provided by the manufacturer.

Preparing a PostgreSQL database

This section describes how to prepare your PostgreSQL database for SourceForge installation. SourceForge is compatible with PostgreSQL 8.2.

For PostgreSQL installation instructions, visit http://www.postgresql.org/docs/

After installing PostgreSQL, do the following:

1. Create the PostgreSQL user as follows:
   
   As root:
   ```
   su - postgres
   createuser -P --createdb --no-createrole <username>
   ```
   Substitute the database username you will be using for `<username>`.

2. Create the database as follows:
   ```
   createdb -E UNICODE -O <username> <database name>
   ```
Preparing an Oracle database

This section describes how to prepare your Oracle database for SourceForge installation. SourceForge is compatible with Oracle 9.2.0.6 with the Oracle 10g (10.2.0.1.0) client. You must upgrade your Oracle client to 10.2.0.1.0 and ensure that the permissions are correct as specified in this section.

For complete Oracle installation instructions, see the Oracle user documentation.

To prepare the Oracle 10g client

1. Install or upgrade your Oracle client to 10.2.0.1.0. Use Oracle 10gR2 Standard Edition. Do not use 10g Express Edition.
2. Do all of the following:
   - export ORACLE_HOME=<oracle_install_directory>
   - chmod 755 ~oracle $ORACLE_HOME
   - cd $ORACLE_HOME
   - find . -type f | xargs chmod o+r
   - find . -type d | xargs chmod o+rx
3. Edit or update $ORACLE_HOME/network/admin/tnsnames.ora to point to your Oracle database. Talk to your Oracle DBA for the proper values.
4. Do all the following to validate your setup:
   - su - sf-admin
   - export ORACLE_HOME=<oracle_install_directory>
   - $ORACLE_HOME/bin/sqlplus <user>/<password>@<db>

You should now be able to connect to the database as sf-admin. If not, you should resolve this before continuing.

To prepare an Oracle database

1. Verify your database uses UTF8 or AL32UTF8 encoding. CollabNet strongly recommends UTF8 as there are known issues with AL32UTF8 in Oracle9 when entering long text values in a user-defined field. Please reference artf21012 when reporting this issue to VA Support.
   To verify your database encoding, use the following query:
   ```sql
   sqlplus <dbadminuser>/<password>@<db> as sysdba
   SQL> SELECT value FROM NLS_DATABASE_PARAMETERS where parameter='NLS_CHARACTERSET';
   ```
2. Create the dbuser/password that you will use to connect from SourceForge to Oracle.
   ```sql
   sqlplus <adminuser>/<password>@<db> as sysdba
   SQL> create user <sf user> identified by <sf passwd> default tablespace <your tablespace> temporary tablespace <temporary tablespace>;
   User created.
   ```
Preparing the database

3. Grant permissions to the user that you just created.

   SQL> grant alter session to <sf user>;
   SQL> grant create table to <sf user>;
   SQL> grant create session to <sf user>;
   SQL> grant unlimited tablespace to <sf user>;
   SQL> exit

Note - The tables and default values will be created by the SourceForge installer.
Preparing the SourceForge server

This section describes how to prepare the SourceForge server for installation.

Preparing the SourceForge server on RedHat or SUSE Linux

SourceForge Enterprise Edition 4.4 runs on a number of Linux versions.

- See “Hardware and software requirements” on page 2 for operating system compatibility.
- Refer to the appropriate documentation for complete installation instructions.

All RedHat Linux documentation is available at http://www.redhat.com/docs/


To prepare the SourceForge server

1. Install the Java 1.5.0_12 SDK.
   You can download the SDK from http://java.sun.com/j2se/1.5.0/download.html
   Download the JDK to the /tmp directory.

   2. Change into the directory where you want to install the Java SDK and run the installation program.

   ```
   cd /usr/local
   sh /tmp/jdk-1_5_0_12-linux-i586.bin
   ```

   This creates a directory called jdk1.5.0_12.

3. Create a link for /usr/local/java

   ```
   ln -s /usr/local/jdk1.5.0_12 /usr/local/java
   ```

   The directory /usr/local/java is referred to as the Java Home directory.

4. For RedHat Enterprise Linux 4 only:

   Install from RedHat ES4.0-U1 RPMS:
   - compat-glibc-2.3.2-95.30
   - compat-glibc-headers-2.3.2-95.30.i386.rpm
   - glibc-kernheaders-2.4-9.1.87.i386.rpm

5. If your configuration uses Apache to serve static files, you will need to update them:

   ```
   # Unzip static-html_4.4.zip into /tmp
   unzip -d /tmp static-html_4.4.zip
   cd /var/www/html
   rm -rf css sf-help sf-images
   unzip /tmp/css_4.4.war -d css -x '*-INF/*'
   unzip /tmp/sf-help_4.4.war -d sf-help -x '*-INF/*'
   unzip /tmp/sf-images_4.4.war -d sf-images -x '*-INF/*'
   ```
6. Copy and unzip the SourceForge installation package to any location on the server. This will be either sf_jboss_postgres_4.4.zip or sf_jboss_oracle_4.4.zip, depending on the database that you use.

    unzip <INSTALLER>.zip

This location is referred to as the SourceForge installation directory.

**Note.** The server that runs SourceForge needs an entry for its hostname in /etc/hosts. This entry cannot be for the loopback address (127.0.0.1). If you see the following line in /etc/hosts:

```
127.0.0.1 <hostname> localhost.localdomain localhost
```

Remove the `<hostname>` so the result looks like:

```
127.0.0.1 localhost.localdomain localhost
```

Add a proper entry for the hostname.
Preparing the SCM servers

Before running the SourceForge SCM integration installer, you must prepare your SCM servers according to the following instructions for the SCM(s) to be supported. Also, if you are adding support for another SCM type after install of the integration server, you must do preparation specific to the SCM type before running configureSCM.sh to add the new SCM.

Installing third party packages

Before you prepare your SCM servers, you must download and install the appropriate third party packages for the operating system on which your SCM server is running. For some operating systems, these packages are in the operating system install media, but not in the base install.

Preparing a SourceForge SCM Integration Server

The following instructions applies to all SourceForge SCM Integration Servers.

1. From sfee.open.collab.net, download the following Python SOAP library rpms for your python version. To find your python version, type:

   python -V

   For Python2.2, download:

   charmod-1.0-1.noarch.python22.rpm
   fpconst-0.7.0-1.vasnosw-py22.rpm
   SOAPpy-0.12.0-1.vasw-py22.rpm

2. For Python2.3, download:

   charmod-1.0-1.noarch.python23.rpm
   fpconst-0.7.0-1.vasw-py23.rpm
   SOAPpy-0.12.0-1.vasw-py23.rpm

   Install the rpms using “rpm -ihv”

3. Create a python2 symlink that points to python if the symlink does not already exist:

   ln -s /usr/bin/python /usr/bin/python2
Preparing a CVS server

To prepare a CVS server, follow the steps in “Preparing the SourceForge server” on page 11 and the steps in ”Preparing the SourceForge SCM Integration server” on page 12 followed by the steps below:

**Note** - CVS servers must be using file based authentication to be integrated with SourceForge. NIS/NIS+ integrations are not possible. Automount, as well as external authentication, may also cause issues and need to be disabled.

1. Verify that CVS version 1.11.x or above is installed and that the “cvs” command is in the root user’s path by logging into the SCM server as the root user and typing:
   
   ```
   cvs -v
   ```

2. Verify that RCS is installed and that the “rcs” command is in the root user’s path by logging into the SCM server as the root user and typing:

   ```
   rcs -V
   ```

3. For SuSE Linux, install the “expect” rpm by logging into the SCM server as the root user and typing:

   ```
   yast2 -i expect
   ```

4. In `install-saturn.conf`, set the following:

   ```
   SOURCEFORGE_INSTALL_CVS=true
   ```
Configuring LDAP for a CVS Integration Server

Authentication on a CVS integration server is performed via UNIX users and groups by default. However, the manipulation of UNIX users and groups, particularly for a large number of users and repositories, can be slow. By configuring the CVS integration server to use LDAP for managing the users and groups, you can significantly speed up the time to create, manipulate, and synchronize users and groups.

**Note** - Do not use this technique to connect a CVS integration to a pre-existing corporate LDAP system.

This approach uses a local, private LDAP server to replace the use of /etc/passwd and /etc/group for user and group management, for the purpose of performance improvement.

The following changes must be performed on every CVS integration server which you want to convert to LDAP:

1. For RHES, ensure that the following RPMs are installed:
   - openldap
   - openldap-clients
   - openldap-servers
   Verify that /etc/openldap/cacerts exists. In RHEL 4, installation of openldap may not create this directory, which is required by openldap to start.

2. Edit /etc/openldap/slapd.conf and change the following values:
   - suffix "dc=sourceforge,dc=com"
   - rootdn "cn=Admin,dc=sourceforge,dc=com"
   - rootpw sfee
   - sizelimit 30000
   You can supply an encrypted password instead of clear text for the password. To do this, execute the command:
     
     slappasswd
     
     You will be prompted for a password, then re-prompted, and a string that looks like the following will be displayed:
     
     (SSHA) 7hc2h50e2oat6rl3havxyy1ljrzyb2
     
     Use that string instead of “sfee” used in the previous rootpw example. The file contents should resemble:
     
     rootpw (SSHA) 7hc2h50e2oat6rl3havxyy1ljrzyb2

3. Configure the LDAP service to start on boot by typing:

   `chkconfig --level 345 ldap on`

4. Make sure the ldap database directory is clean and has the correct permissions:
Preparing the SCM servers

```
rm -f /var/lib/ldap/*
chown -R ldap.ldap /var/lib/ldap
```

5. Start the LDAP server by typing:

```
/etc/init.d/ldap start
```

6. Create an initial LDIF (LDAP Interchange Format) file for your groups. It is important that all lines in the ldif content do not start with white space. Refer to the following for ldif content:

```
dn: dc=sourceforge,dc=com
  dc: sourceforge
  objectClass: top
  objectClass: domain

dn: ou=Users,dc=sourceforge,dc=com
  ou: Users
  objectClass: organizationalUnit

dn: ou=Groups,dc=sourceforge,dc=com
  ou: Groups
  objectClass: organizationalUnit

dn: cn=sfee,ou=Groups,dc=sourceforge,dc=com
  cn: sfee
  objectClass posixGroup
  gidNumber: 30000
```

7. Add the ldif information with the following command:

```
ldapadd -x -D"cn=Admin,dc=sourceforge,dc=com" -W < initial.ldif
```

When you are prompted for a password, type the password you used in slapd.conf (in this example, sfee).

8. Manually remove SFEE-created groups and users from /etc/group and /etc/passwd. These are usually grouped at the end of the respective files. SFEE-created groups include sfall, sfunrest, and all groups names “reps” with a numeric suffix (for example: reps1001).

If users are left in the /etc/passwd or /etc/shadow files, permission errors may result for this users when committing code to repositories on that integration server.

If SFEE-created groups are left in the /etc/group and /etc/gshadow files, users of those groups may experience permission errors when checking in. If groups are removed from these files, a synchronize external system call will be required to restore correct permissions.
9. Configure the server to authenticate from LDAP.
   On RHES this is performed using the ‘authconfig’ command. If you have a valid display defined, this command will pop up an X window; otherwise you can use the curses interface.
   • Check the ‘Use LDAP’ checkbox.
   • Specify the base DN and server. Do not check ‘Use TLS’.
   • Click ‘Next.
   • Check ‘Use LDAP Authentication’. Fill in the DlDAP basedn and host information.
   • Click ‘OK’.
   You are returned to the filesystem. This program modifies some /etc/pam.d entries and writes out a valid /etc/ldap.conf.

10. Restart sshd with the following command:
    
    /etc/init.d/sshd restart

11. Verify configuration.
    In /etc/nsswitch.conf, be sure that the passws, shadow, and group entries look like the following:
    • passwd: files ldap
    • shadow: files ldap
    • group: files ldap
    Verify that /etc/ldap.conf contains the following values:
    • nss_base_passwd ou=Users,dc=sourceforge,dc=com?one
    • nss_base_shadow ou=Users,dc=sourceforge,dc=com?one
    • nss_base_group ou=Groups,dc=sourceforge,dc=com?one
    Log in now. If you cannot, verify that /etc/ldap.conf contains the following entries:
    • pam_filter objectClass=posixAccount
    • pam_login_attribute uid
12. Configure SourceForge integration server to update LDAP.
On the SourceForge integration server, edit /etc/sourceforge.properties, and add the following entries:

<table>
<thead>
<tr>
<th>Entry</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>sfmain.ldap.host</td>
<td>The host of the LDAP server. This should be localhost.</td>
</tr>
<tr>
<td>sfmain.ldap.port</td>
<td>The port of the LDAP server. This should be 389.</td>
</tr>
<tr>
<td>sfmain.ldap.users.container</td>
<td>Schema address that users are added into (for example: ou=Users,dc=sourceforge,dc=com)</td>
</tr>
<tr>
<td>sfmain.ldap.groups.container</td>
<td>Schema address that groups are added into (for example: ou=Groups,dc=sourceforge,dc=com)</td>
</tr>
<tr>
<td>sfmain.ldap.users.startid</td>
<td>The numeric userid to start counting from when creating new users. This should be 30000.</td>
</tr>
<tr>
<td>sfmain.ldap.groups.startid</td>
<td>The numeric groupid to start counting from when creating new groups. This should be 30005.</td>
</tr>
<tr>
<td>sfmain.ldap.bind.dn</td>
<td>The “root dn” for the LDAP server. This must be the value specified in slapd.conf (for example: cn=Admin,dc=sourceforge,dc=com).</td>
</tr>
<tr>
<td>sfmain.ldap.bind.password</td>
<td>The password for the root dn.</td>
</tr>
</tbody>
</table>

- Next, change the sfmain.integration.os from linux to linux_ldap and change the sfmain.integration.user_group from users to sfee.
- Restart the integration server when you have completed all changes to /etc/sourceforge.properties by executing the following command:
  
  ```bash
  /etc/init.d/sourceforge-integration restart
  ```

13. Navigate to the Integration Systems page and synchronize permissions for all managed CVS integration servers.
14. When synchronize permissions has completed, you must correct the permissions on the home directories of your users. On each CVS server that you have converted to LDAP, perform the following:

```bash
    cd /home
    for i in *
        do
            chown -R $i.root $i
    done
```

You may see some errors; this is normal and indicates disabled/deleted users.

15. Running synchronize permissions will send email to your entire user community. This occurs because the users need to click on the link in the email to set their LDAP password (by entering their current SFEE password). The password cannot be set automatically during migration because only the encrypted version is available.
Preparing a Subversion server

SourceForge has been fully tested with the following Subversion versions:

- Subversion 1.2.3 (FSFS) on RedHat Enterprise Server 3
- Subversion 1.3.1 (FSFS) on SUSE Linux Enterprise Server 9
- Subversion 1.4.3 (FSFS) on RedHat Enterprise Linux 4
- Subversion 1.4.3 (FSFS) on CentOS4

For additional information on installing and configuring Subversion, see:
http://svnbook.red-bean.com/

To prepare a Subversion server, follow the steps in “Preparing the SourceForge server” on page 11 and the steps in “Preparing the SourceForge SCM Integration server” on page 12 followed by the steps below:

Configure the SourceForge installer to install Subversion integration

1. Edit install-saturn.conf and set the following:
   
   SOURCEFORGE_INSTALL_SUBVERSION=true
   ...
   SUBVERSION_URI=/svn/repos
   SUBVERSION_REPOSITORY_BASE=/svnroot
   INTEGRATION_SERVER_HOST=<Subversion server name>

2. In install-saturn.conf, leave the SUBVERSION_FSTYPE as fsfs.

Installing Subversion for Red Hat Linux

1. Copy or download all of the files for Subversion into a new directory, but do not install them yet.

   A SourceForge Subversion server requires the following RPMs:

   Subversion 1.2.3 on RHES3:
   
   - subversion-1.2.3-1
   - subversion-python-1.2.3-1
   - mod_dav_svn-1.2.3-1

   The above RPMs are available at:
   http://sfee.open.collab.net/sffrs/do/listReleases/projects.sfdl/frs.3rd_party_components
   
   - neon-0.24.7-1

   This RPM is available on the RedHat installation media.
Preparing for Installation

Preparing the SCM servers

Subversion 1.4.3 on RHEL4:
- subversion-1.4.3-1
- subversion-python-1.4.3-1
- swig-1.3.25-1
- mod_dav_svn-1.4.3-1
- apr-0.9.12-2
- apr-util-0.9.12-1
The above RPMs are available at:
http://sfee.open.collab.net/sf/frs/do/listReleases/projects.sfdl/frs.3rd_party_components
- neon-0.24.7-4
This RPM is available on the RedHat installation media.

2. For Red Hat ES3/AS3, install latest versions provided by Red Hat for the following RPMs:
- httpd
- mod_ssl
Note: If you had a different version of httpd already installed, backup your config files httpd.conf, ssl.conf and any other customization that you had made.

3. Install mod_python RPM for your Apache version

4. Before installing the Subversion RPMs:
- If RedHat-config-httpd is installed on the Subversion server, you may not be able to start Apache after configuring the server for SourceForge.
  Uninstall it using:
  ```
  rpm -e RedHat-config-httpd
  ```
- If db4-utils is installed on the Subversion server, uninstall it using:
  ```
  rpm -e db4-utils
  ```

5. Install the Subversion RPMs:
To install the RPMs, use `rpm -Uvh *.rpm` in the local downloaded directory. The RPMs may have dependencies. These are available on the RedHat install media.

6. Verify load of DAV module
After installing the RPMs, verify that the following line is in your Apache /etc/httpd/conf/httpd.conf file is not commented out:

```
LoadModule dav_module modules/mod_dav.so
```
Preparing the SCM servers

Installing Subversion for SuSE Linux Enterprise Server 9

1. Build and install subversion 1.3.1 from source code by following the instructions below:

   **Install the following RPMs using Yast2b**
   - apache2-devel
   - python-devel

   **Download the subversion source by typing**
   ```
   wget http://subversion.tigris.org/downloads/subversion-1.3.1.tar.bz2
   export http_proxy=http://<proxy_host>:<proxy_port>
   ```

   **Untar the subversion source by typing**
   ```
   tar -jxvf subversion-1.3.1.tar.bz2
   ```

   **Build and test subversion 1.3.1 by typing**
   ```
   cd subversion-1.3.1
   ./configure --enable-dav --enable-so --with-apxs=/usr/sbin/apxs2 | tee configure.log
   make all swig-py check | tee make.log
   ```

   **Install subversion 1.3.1 by typing**
   ```
   make -k install | tee -a make.log
   make install-swig-py | tee -a make.log
   echo /usr/local/lib/svn-python > /usr/lib/python2.3/site-packages/subversion.pth
   ln -s /usr/local/lib/libsvn_* /usr/lib
   ```

2. Install using yast2:
   - apache2-mod_python
   - python-xml

Preparing a Perforce server

To prepare a Perforce server, follow the steps in “Preparing the SourceForge server” on page 11 and the steps in “Preparing the SourceForge SCM Integration server” on page 12 followed by the steps below:

1. Install Perforce version 2006.2. For information on installing Perforce, see the following URL:
   ```
   http://www.perforce.com/perforce/technical.html
   ```

Configure the SourceForge installer to install Subversion integration

In `install-saturn.conf`, set the following:

```python
SOURCEFORGE_INSTALL_PERFORCE=true
...`

PERFORCE_PORT=localhost:1666
INTEGRATION_SERVER_HOST=<Perforce server name>

The parameter PERFORCE_PORT must be set to <hostname>:1666. The only valid values for the hostname are localhost or the hostname of the server on which Perforce and the SourceForge integration adapter are installed. Perforce and the SourceForge integration adapter must be installed on the same server.
Editing the SourceForge install configuration file

The SourceForge Enterprise Edition installer is controlled by the configuration file

```
<SOURCEFORGE_INSTALLER_DIR>/install-saturn.conf
```

Before installation, you need to edit the file and comment out the line DIE_IMMEDIATELY=yes and then set any parameters for your installation, like specifying the components to be installed.

If you wish to install the SourceForge Application Server, set SOURCEFORGE_INSTALL_STANDALONE to true. You will also need to set the DATABASE variables to point to the database you have created for SourceForge. **Finally, be sure to set the JAMES_ADMIN_PASSWORD to something other than the default password.**

If you wish to also install SCM servers, you can set one or more of the following variables to true: SOURCEFORGE_INSTALL_CVS, SOURCEFORGE_INSTALL_SUBVERSION, and SOURCEFORGE_INSTALL_PERFORCE. There are other configuration options for each SCM server.

Global parameters

You must edit the parameters listed in the following table.

<table>
<thead>
<tr>
<th>Parameter Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIE_IMMEDIATELY</td>
<td>Either comment this variable out or set it to any value other than 'yes'.</td>
</tr>
<tr>
<td>SOURCEFORGE_INSTALL_STANDALONE</td>
<td>Set each value to true or false based on the SourceForge and/or SCM servers that you are installing.</td>
</tr>
<tr>
<td>SOURCEFORGE_INSTALL_CVS</td>
<td>[ ]</td>
</tr>
<tr>
<td>SOURCEFORGE_INSTALL_SUBVERSION</td>
<td>[ ]</td>
</tr>
<tr>
<td>SOURCEFORGE_INSTALL_PERFORCE</td>
<td>[ ]</td>
</tr>
<tr>
<td>SOURCEFORGE_INSTALL_DIR</td>
<td>The directory where SourceForge and all of its components will be installed.</td>
</tr>
<tr>
<td>CREATE_DATABASE_SCHEMA</td>
<td>Set to ‘yes’ on first installation. (See “Failed installation” on page 30.)</td>
</tr>
<tr>
<td>DATABASE_TYPE</td>
<td>postgres (for PostgreSQL installations)</td>
</tr>
<tr>
<td></td>
<td>oracle (for Oracle installations)</td>
</tr>
<tr>
<td>DATABASE_USERNAME</td>
<td>The username that you use to connect to the database. (The database user.)</td>
</tr>
<tr>
<td>DATABASE_PASSWORD</td>
<td>The database user’s password.</td>
</tr>
<tr>
<td>DATABASE_HOST</td>
<td>The database host name.</td>
</tr>
<tr>
<td>Parameter Value</td>
<td>Value</td>
</tr>
<tr>
<td>----------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>DATABASE_PORT</td>
<td>The database port.</td>
</tr>
<tr>
<td><strong>For PostgreSQL installations</strong> - The default port for PostgreSQL is 5432. If you installed your PostgreSQL server on a different port, change this value accordingly.</td>
<td></td>
</tr>
<tr>
<td>DATABASE_NAME</td>
<td>The database name.</td>
</tr>
<tr>
<td>JAVA_HOME</td>
<td>/usr/local/java</td>
</tr>
<tr>
<td>JBOSS_USER</td>
<td>The SourceForge user.</td>
</tr>
<tr>
<td>SOURCEFORGE_VAR_DIR</td>
<td>The directory containing SourceForge data files.</td>
</tr>
<tr>
<td>SOURCEFORGE_LOG_DIR</td>
<td>The directory containing SourceForge log files.</td>
</tr>
<tr>
<td>SOURCEFORGE_APPSERVER_HOST</td>
<td>The host name of the SourceForge server</td>
</tr>
<tr>
<td>SOURCEFORGE_ROOT_URL¹</td>
<td>The base URL that web clients will use to access SourceForge.</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong>: You must leave the /sf directory at the end of this URL.</td>
</tr>
<tr>
<td>SOURCEFORGE_SYSTEM_EMAIL</td>
<td>The e-mail address of the user that will receive system e-mails. Cannot be accounts on the SFEE server itself.</td>
</tr>
<tr>
<td>SOURCEFORGE_ADMIN_EMAIL</td>
<td>The e-mail address of the SourceForge administrator. Cannot be accounts on the SFEE server itself.</td>
</tr>
<tr>
<td>JAMES_DNS_HOST</td>
<td>The address of a DNS server that can be used from the SourceForge server.</td>
</tr>
<tr>
<td>JAMES_MTA_HOST</td>
<td>This is the host that runs the SourceForge James server. Typically this is the same host as the SOURCEFORGE_APPSERVER_HOST.</td>
</tr>
<tr>
<td>JAMES_GATEWAY_HOST</td>
<td>This is the MTA used for delivering email if the JAMES_MTA_HOST is unable to deliver email directly. For example, if it is behind a firewall and is unable to connect to external mail hosts.</td>
</tr>
<tr>
<td>JAMES_ADMIN_PASSWORD</td>
<td>Password to use for the JAMES admin user. This should be set to a strong password for security.</td>
</tr>
</tbody>
</table>
Editing the SourceForge install configuration file

<table>
<thead>
<tr>
<th>Parameter Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>JAMES_ACCEPTED_RELAYS</td>
<td>This is a comma-separated list of wildcard-matched hostnames that are allowed to relay mail through the SourceForge JAMES MTA.</td>
</tr>
<tr>
<td></td>
<td>With the exception of localhost, if a host is not in the list of accepted relays, mail from that host will be spooled to the var/mail/spam mail folder under <code>&lt;SOURCEFORGE_INSTALL_DIR&gt;/james/james-2.1.2/apps/james/var/mail/spam</code></td>
</tr>
<tr>
<td></td>
<td>For additional information on JAMES configuration, see: <a href="http://james.apache.org/configuration_v2_0.html">http://james.apache.org/configuration_v2_0.html</a></td>
</tr>
<tr>
<td>MAIL_EXCHANGE_HOST</td>
<td>This variable is used by the application server to determine where the SourceForge James server is located. The default value of localhost is generally sufficient. If desired, you can point to a different MTA than the James server, since SourceForge does not do outbound email post-processing.</td>
</tr>
</tbody>
</table>

1 If you intend to set up Apache with your SourceForge installation, set the SourceForge root URL to an outside URL, without any port number.

**Note** - The directories where the SourceForge data (SOURCEFORGE_VAR_DIR) and logs (SOURCEFORGE_LOG_DIR) are stored should be on a large partition. The contents of these directories will grow and will contain increasing amounts of SourceForge data. Back up these directories frequently.
**Internationalization parameters**

You must edit the variables in the following table only if you are going to use a non-English language.

<table>
<thead>
<tr>
<th>Parameter Value</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SOURCEFORGE_MULTIPLE_LANGUAGE_ENABLED</td>
<td>Set this option to &quot;true&quot; if you want users to be able to select their preferred language.</td>
</tr>
<tr>
<td>SOURCEFORGE_DEFAULT_LOCALE</td>
<td>If users do not select a preferred language, their language defaults to this value.</td>
</tr>
<tr>
<td>SOURCEFORGE_DEFAULT_EXPORT_ENCODING</td>
<td>This is the encoding to use when exporting SourceForge data.</td>
</tr>
<tr>
<td>MAIL_CHARACTER_ENCODING_ASCII</td>
<td>This specifies the &quot;charset&quot; portion of outgoing email content-type header for single-byte email.</td>
</tr>
<tr>
<td>MAIL_CHARACTER_ENCODING_MB</td>
<td>This specifies the &quot;charset&quot; portion of outgoing email content-type header for multiple-byte email</td>
</tr>
<tr>
<td>MAIL_TRANSFER_ENCODING_ASCII</td>
<td>This is the value to use with the content-transfer-encoding header for single-byte email.</td>
</tr>
<tr>
<td>MAIL_TRANSFER_ENCODING_MB</td>
<td>This is the value to use with the content-transfer-encoding header for multiple-byte email.</td>
</tr>
</tbody>
</table>
Editing the SourceForge install configuration file
The SourceForge Enterprise Edition installer has been updated to allow you to install SourceForge and your SCM servers together or separately, on one or more physical servers. Before running the SourceForge installer, you must prepare your SourceForge and SCM servers for SourceForge installation. See *Chapter 1, Preparing for Installation* on page 1.

After preparing your servers for SourceForge installation, you can then run the installer to install the desired components.

After completing your SourceForge installation, you can then configure Apache, configure LDAP integration, and add additional SCM servers.

This chapter contains the following information:

- “Running the installation program” on page 30
- “Finishing the SourceForge installation” on page 32
- “Finishing SCM installation” on page 33
- “/etc/httpd/conf/httpd.conf” on page 71.
- “Configuring LDAP integration” on page 34
- “Adding additional SCM servers” on page 38
Running the installation program

You must be logged in as root to install SourceForge.

1. Change directory into the unzipped installer directory:

   \texttt{cd <INSTALLER>}

2. Check the install parameters in \texttt{install-saturn.conf}.

3. Run the installation program:

   \texttt{export JAVA_HOME=/usr/local/java}
   
   \texttt{bash ./install-saturn.sh install-saturn.conf}
   
   - If the installation program completes successfully, you will see the following line at the end of
     your output:
     \texttt{Installation Process Complete}
   
   - If the installation program does not complete successfully, see the following Failed Installation
     section.

Failed installation

If the installation process fails, you will see error messages and the installer will clean up the installed
files. You can see a full output log in \texttt{/tmp/sourceforge_install.log}, or whatever location you
specified in \texttt{SOURCEFORGE_INSTALLATION_LOG}.

From the information in the installation log file you should be able to correct your settings.

If the database scheme was created successfully, as indicated in the installation log file, you must set the
variable \texttt{CREATE_DATABASE_SCHEMA} to 'no' in order to run the installer again.

Run the installer again. If it still does not succeed, check the installation log file again and correct the
configuration file.

Following are some potential reasons for a failed installation, and the steps you can take to correct them.

<table>
<thead>
<tr>
<th>Error message</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>DATABASE_TYPE must be one of:</td>
<td>Change the DATABASE_TYPE option in the install-saturn.conf to be either oracle or postgres.</td>
</tr>
<tr>
<td>oracle postgres.</td>
<td></td>
</tr>
<tr>
<td>Installation was aborted by user.</td>
<td>Change the DIE_IMMEDIATELY option in the install-saturn.conf file to &quot;no&quot;. This option was placed in the configuration file to force customization of the configuration file.</td>
</tr>
<tr>
<td>Variable XXX must be set.</td>
<td>This is a catch-all error where XXX represents an option in the install-saturn.conf file. Verify that this option exists and is not commented out.</td>
</tr>
</tbody>
</table>
### Error message | Solution
--- | ---
Local user XXX must exist. | A local system user that is required by SourceForge does not exist. Either create the required user, or change the corresponding variable in the `install-saturn.conf` file to a user that already exists.
Local group XXX must exist. | A local system group that is required by SourceForge does not exist. Either create the group or change the corresponding variable in the `install-saturn.conf` file to refer to an existing group.
XXX has a bad shell. | The local user specified has a shell which denies logins such as `/sbin/nologin` or `/bin/false`. Either change the login shell of that user to be a valid login shell, or specify a user who does have a valid login shell.
XXX must be in path. | The executable is required by either SourceForge or the SourceForge installer to properly function. If the executable in question does exist on the server, add the directory of the executable to PATH.
Unsupported operating system: XXX. | Verify that the host being installed on is a supported platform. See “Hardware and software requirements” on page 2.
Check Port Available FAILED: Port XXX on XXX is not available. | SourceForge verifies that required ports are available at installation time. If one or more of these are already in use, the installation fails. If this happens, disable the existing service on the server that uses the same port.
Finishing the SourceForge installation

After you have run the installer to install SourceForge, you must complete the following steps to finish the installation.

The main SourceForge process will run under the SourceForge user. Some processes have to run as the root user.

1. Configure your session timeout limit by editing the session-timeout parameter in:

   `<SOURCEFORGE_INSTALL_DIR>/jboss/jboss-3.2.6/server/default/deploy/jbossweb-tomcat50.sar/conf/web.xml`

   The default session timeout for the version of JBoss shipped with SourceForge Enterprise Edition 4.4 is 30 minutes.

2. As root, start SourceForge with:

   `/etc/init.d/sourceforge-standalone start`

   You should see the following output:

   Using PHOENIX_HOME:   /usr/local/sourceforge/james/james-2.1.2
   Using PHOENIX_TMPDIR: /usr/local/sourceforge/james/james-2.1.2/temp
   Using JAVA_HOME:      /usr/local/java
   Starting Phoenix:
   Phoenix running pid=21188

   If you point your browser to `http://<SOURCEFORGE_APPSERVER_HOST>:8080/sf` you should see the SourceForge home page.

3. To configure Linux to start and stop SourceForge on system boot and shutdown

   For the SourceForge application:

   `chkconfig --add sourceforge-standalone`

   For the SourceForge integration:

   `chkconfig --add sourceforge-integration`

   For the Apache Server:

   `chkconfig --add httpd # Red Hat Linux`
   `chkconfig --add apache2 # SuSE Linux`

Configuring Apache

To properly configure your Apache install to work with SourceForge, please read Appendix “/etc/httpd/conf/httpd.conf” on page 71.
Finishing SCM installation

After you have run the installer to install one or more CVS, Subversion, or Perforce servers, you must complete the following steps to finish the installation.

1. Start SourceForge integration server.
   ```bash
   /etc/init.d/sourceforge-integration start
   ```

2. Configure Linux to start and stop the SourceForge integration server on boot and shutdown:
   ```bash
   chkconfig --add sourceforge-integration
   ```

In addition, for Subversion, you will need to restart Apache:

1. Restart Apache.
   ```bash
   /etc/init.d/httpd restart # For Red Hat Linux
   /etc/init.d/apache2 restart # For SuSE Linux
   ```

2. Configure Linux to start and stop Apache on system boot and shutdown:
   ```bash
   chkconfig --add httpd # For Red Hat Linux
   chkconfig --add apache2 # For SuSE Linux
   ```

Configuring Apache

If you are running SourceForge SCM integration server on a different machine than the SourceForge Application server and there is a firewall between the two machines, you must either configure your firewall to allow access for the SourceForge Application Server to connect to port 7080 of the SourceForge integration server or configure Apache on the integration server to proxy port 80 to port 7080 by adding the following lines the Apache `httpd.conf`:

```bash
RewriteEngine on
ProxyPassReverse / http://localhost:7080/
```

If you intend to run your Apache server with https support, you must add these lines to the SSL section of the `httpd.conf` file. Otherwise you can add them to the end of the file.
Configuring LDAP integration

This section provides instructions on configuring your SourceForge JBoss installation to authenticate against your corporate LDAP server. LDAP authentication is optional. You can use either SourceForge authentication or LDAP authentication, but you cannot use both with a single SourceForge installation.

If you configure SourceForge to authenticate against an LDAP server, all SourceForge authentication will be done against the LDAP server. If the LDAP server is down, all SourceForge authentication will be disabled until the LDAP server is restored. If a user does not exist on the LDAP server, or is deleted from the server, he or she will not be able to log in to SourceForge. The admin user is the only exception to this rule.

SCM authentication is not managed by LDAP, but each SourceForge user’s SCM password is synchronized automatically with his or her LDAP password upon logging in to SourceForge.

**Note about existing users** - If your SourceForge installation has user accounts that were created before LDAP integration was enabled, these user accounts will remain active. However, the users will not be able to log in to SourceForge unless their usernames in SourceForge exactly match their usernames on the LDAP server. If so, they will be able to log in to SourceForge using their LDAP passwords.

Before you begin

Before you begin converting your JBoss SourceForge installation to authenticate against an LDAP server, you must do the following:

1. Shut down SourceForge.
2. Add the following property to the <SOURCEFORGE_INSTALL_DIR>/sourceforge_home/etc/sourceforge_configuration.properties file:
   
   `sf.useExternalUserAuthentication=true`

3. In the same sourceforge_configuration.properties file, make sure that the system.password.min-length property is set, and that it matches the limit used on the LDAP server.
   
   If your LDAP server does not enforce a minimum password length, set the system.password.min-length property to zero.
   
   If a password is used in LDAP that is shorter than the minimum allowable password length in SourceForge, you will not be able to create the user in SourceForge.
Configuring SourceForge for LDAP authentication

The configuration file that controls user authentication is:

```
<SOURCEFORGE_INSTALL_DIR>/jboss/jboss-3.2.6/server/default/conf/login-config.xml
```

You must modify the application-policy block of this file to enable SourceForge to authenticate against your LDAP server. To modify the application-policy block successfully, you must do the following:

1. Replace the default application-policy block with the code listed in “Example application-policy block” on page 36.
   
   The application-policy block begins on line 100 of the login-config.xml file.

2. In the new application-policy block, modify the option values listed in “Required option value modifications” on page 37.

3. Restart SourceForge.

Default application-policy block

By default, the application-policy block of the login-config.xml file looks like this:

```
<application-policy name="SourceForge">
  <authentication>
    <login-module
code="com.vasoftware.sf.server.apps.sfmain.authentication.UsernamePasswordInDat
abaseLoginModule"flag="required">
      <module-option name="DataSourceName">java:/SourceForgeDS</module-option>
    </login-module>
  </authentication>
</application-policy>
```

You must replace it with the code in “Example application-policy block” on page 36.
Configuring LDAP integration

Example application-policy block

Replace the default application-policy block of the login-config.xml file with the following code. Option values that must be modified are highlighted in **bold**.

```xml
<application-policy name="SourceForge">
  <authentication>
    <login-module code="org.jboss.security.auth.spi.LdapLoginModule" flag="sufficient">
      <module-option name="allowEmptyPasswords">false</module-option>
      <module-option name="principalDNPrefix">uid=</module-option>
      <module-option name="principalDNSuffix">,ou=People,dc=dev,dc=sf,dc=net</module-option>
      <module-option name="java.naming.factory.initial">com.sun.jndi.ldap.LdapCtxFactory</module-option>
      <module-option name="java.naming.provider.url">ldap://util.dev.sf.net:389/</module-option>
      <module-option name="java.naming.security.authentication">simple</module-option>
    </login-module>
  </authentication>
</application-policy>
```

**Note** - This example contains a single login-module section. If you are authenticating against multiple LDAP servers, include one login-module section per LDAP server, with the required option values modified appropriately for each one. If the same username exists in more than one LDAP server, the instance on the first LDAP server will be used.

After you have replaced the default application-policy block with the code in this section, you must make the modifications specified in “Required option value modifications” on page 37.
Required option value modifications

To enable SourceForge to authenticate against your LDAP server, you must modify the following option values in the application-policy block of the login-config.xml file.

1. When the username is passed to the login module from SourceForge, it is translated into a DN for lookup on the LDAP server. The DN that is sent to the LDAP server is:

   \[
   <\text{principalDNPrefix}><\text{username}><\text{principalDNSuffix}>
   \]

   - principalDNPrefix - Replace principalDNPrefix with your LDAP username parameter. In the example application-policy block, the username is stored in LDAP as the \text{uid} parameter.

   \textbf{Note} - Make sure to include the trailing '=' in the prefix.

   - principalDNSuffix - Replace principalDNSuffix with the LDAP domain in which usernames are stored. In the example application-policy block, the username is stored in the People organizational unit in the dev.sf.net domain. This is represented as:

   \[
   ,\text{ou}=\text{People},\text{dc}=\text{dev},\text{dc}=\text{sf},\text{dc}=\text{net}
   \]

   \textbf{Note} - Make sure to include the leading comma in the suffix if one is needed.

2. java.naming.provider.url - Replace java.naming.provider.url with the URL of your LDAP server.

   In the example application-policy block, the URL of the LDAP server is:

   \[
   \text{ldap://util.dev.sf.net:389/}
   \]

   \textbf{Note} - Make sure to include \text{ldap://} at the beginning of the URL.

Completing your SourceForge configuration

To complete your SourceForge configuration and enable your SourceForge JBoss installation to authenticate against your corporate LDAP server, you must restart SourceForge.
Adding additional SCM servers

If you have already installed SourceForge, either with or without one or more SCM servers, you can install additional SCM servers in one of the following ways:

- To install one or more SCM servers on a physical server on which you have already installed SourceForge, but no SCM servers, run the installer again to install the SCM servers.
  
  See “Installation procedures” on page 6.

- To install an additional SCM server on a physical server on which you have already installed one or more SCM servers, you must run a script that prompts you for the necessary SCM configuration information and installs the SCM server.
  
  See “Running the SCM configuration script” on page 38.

Running the SCM configuration script

To install an additional SCM server on a physical server on which you have already installed one or more SCM servers, you must run a script that prompts you for the necessary SCM configuration information and installs the SCM server. You can install any additional CVS, Subversion, or Perforce SCM server in this way.

**Caution** - Do not run the installer again. This can damage your current SourceForge installation and any SCM servers that you have already installed.

You can find the SCM configuration script at:

```
<SOURCEFORGE_INSTALL_DIR>/sourceforge_home/bin/configureScm.sh
```

**To run the SCM configuration script**

1. Set JAVA_HOME to your Java directory.
2. Run the `configureScm.sh` script.
3. When prompted, enter the SCM type that you want to configure.
   
   Values are “subversion”, “cvs”, or “perforce”.
4. Enter or accept the default values when prompted.

When finished, your SCM server will be integrated with SourceForge
CHAPTER 3

SourceForge Administration

This chapter provides information on administering your SourceForge installation. It includes information on backing up SourceForge, security precautions, operating system maintenance, and CollabNet support.

This chapter contains the following information:

- “Backups and restoration” on page 40
- “Server logs” on page 43
- “SourceForge security” on page 47
- “Firewalling, operating system, and Apache security” on page 53
- “Third party application support limitations” on page 60
- “SourceForge and operating system maintenance” on page 62
- “SourceForge Support resources” on page 63
Backups and restoration

SourceForge stores data in the database and on the file system. This chapter provides instructions on how to back up all data comprehensively so that it can be restored in the event of unrecoverable failures. SourceForge should be shut down prior to backing up any data.

Please note that the items listed in this section address only the data that is either created by or a part of SourceForge. Non-SourceForge specific data, such as operating system-based content, configuration files, and other 3rd party applications will also require a backup and restoration routine to ensure that the entire server can be restored in the event of a catastrophic failure. Please contact your application or operating system vendor for specific guidance on backup strategies for their products.

JBoss

The SourceForge application server stores all files uploaded into SourceForge on the file system. This includes:

- Files uploaded into the File Release System
- Documents uploaded into the Document Manager
- Any other files that are attached to tracker artifacts or discussion forum posts.

The search indices, which are used to provide search results, are also stored on the file system. To comprehensively backup all SourceForge-stored data present on the server, you should back up the `<SOURCEFORGE_VAR_DIR>`.

By default, this is under `<SOURCEFORGE_INSTALL_DIR>/var`.

If you would like to back up the SourceForge logs, you should also back up the `<SOURCEFORGE_LOG_DIR>`. This directory stores the following logs:

- jboss_console.log
- server.log
- vaexternalintegration.log
- vamessages.log
- vadaemon.log

By default, this is under `<SOURCEFORGE_INSTALL_DIR>/log`.

When restoring this data from backup, SourceForge must be shut down first.
CVS

On the integration server, the CVS repository is typically stored in /cvsroot.

When SourceForge users are granted access to one or more SourceForge managed CVS repositories, shell accounts are created for each of them in /etc/passwd. Therefore the files /etc/passwd, /etc/group, /etc/shadow and /etc/gshadow (on Linux only) should be backed up along with the all CVS data. Additionally, all of /home should be backed up, as ssh public keys are stored inside of each users $HOME on the SourceForge CVS server.

Subversion

On the integration server, the Subversion repository is typically stored in /svnroot.

Subversion depends on Apache, so the Apache configuration files /etc/httpd/conf/httpd.conf and /etc/httpd/conf.d/sf_subversion.conf should be backed up.

PostgreSQL backups, restoration, and optimization

All database platforms require regular maintenance to maximize performance and maintain stability. In the case of PostgreSQL, this maintenance is relatively simple and straightforward. There are three basic maintenance categories: backups, restoration, and optimization.

A PostgreSQL database can be backed up safely while online using the native pg_dump command. Note that for this example, the name of the SourceForge database is assumed to be 'sf':

    pg_dump -Ft -b -o sf > sf.tar

In the above example, the database is dumped into a GNU tar formatted file. See the PostgreSQL pg_dump man page for additional information and examples.

SourceForge must be shut down before restoring the database from a dump. Create a database and user with the names used for SourceForge, then restore the database using the native pg_restore command. This example also assumes that the name of the SourceForge database is 'sf':

    createuser -U $SFUSER
    createdb -E UNICODE -U $SFUSER sf
    pg_restore -d sf sf.tar

It may also be necessary to restore ownership of the restored tables to the SFEE database user. Something like the following will work (again, assuming the database is called 'sf'):

    for i in `echo "\d" | psql sf | awk '{print $3}'`
do
        echo "ALTER TABLE $i OWNER TO $SFUSER;" | psql sf
    done
See the PostgreSQL pg_restore man page for additional examples and information.

Normal use of database software often creates data overhead that needs to be cleaned periodically in order to ensure optimal speed and stability. This overhead is usually the result of temporary files and indexes that the database creates (analogous to a fragmented hard disk.) Optimizing a PostgreSQL database means running a built-in utility called "vacuum." This utility runs on a live database and, like the backup command, can be scripted to run nightly during minimal server load.

To vacuum the SF database, run the following command as the PostgreSQL user:

```
vacuumdb -z <database_name> && vacuumdb -z <database_name>
```

PostgreSQL 8 adds an autovacuuming feature, which can be configured to perform this task automatically. See "RedHat postgresql.conf" on page 67 for details.

Please see the following vacuumdb documentation for more information on how vacuum works, and its options:


---

**Oracle**

There are a wide variety of commercial applications available for backing up and restoring Oracle databases. SourceForge has no special Oracle database requirements for performing backups. Thus, you are free to use whatever tools your organization traditionally uses for Oracle database maintenance.

If you do not already have Oracle database best practices in place, it is recommended that you use the native exp command for dumping an Oracle database. Note that for this example, the name (tnsname) of the SourceForge database is assumed to 'sf':

```
exp sfuser/sfpasswd@sf file='sf_DB.dmp' owner=sfuser INDEXES=N
```

The above command will dump the database into the file sf_DB.dmp. Running 'exp -help' will provide additional information on the command syntax and options.

SourceForge must be shut down before backing up or restoring the database. Create a database and user with the names used for SourceForge, then restore the database using the native imp command. This example also assumes that the name of the SourceForge database (tnsname) is 'sf':

```
imp system/system@sf file=sf_DB.dmp fromuser=sfuser touser=sfuser
INDEXES=N STATISTICS=recalculate commit=y
```

Running 'imp -help' will provide additional information on the command syntax and options.

CollabNet strongly recommends that an experienced Oracle DBA be available to perform Oracle-related administration functions.
Server logs

SourceForge relies on several different components, each of which generates its own logs. The logs are useful not only for debugging problems, but also to ensure that the application is performing to expectations.

JBoss

The JBoss application server writes several different logs under the
<SOURCEFORGE_INSTALL_DIR>/log directory.

- boot.log - The boot.log logs the JBoss startup and shut down notifications. This log is overwritten each time JBoss is (re)started.
- localhost_access - The localhost_access log records access to the application from a remote host, similar to the Apache access_log. This log is rotated each day, and the files have a date stamp appended to their name, such as localhost_access2004-11-26.log.
- server.log - The server.log logs all the activities of the application server, including any exceptions. This log is the best place to begin debugging SourceForge server error exception ids (exid).
- session-info.log - The session-info.log records when new sessions are created. This log is overwritten each time JBoss is (re)started.
- vamessages.log - The vamessages.log records SourceForge-specific actions, including some SQL queries that are sent to the backend database. This log is rotated each time it reaches 100MB in size. When rotated the older files have a number appended to the end, such as vamessages.log.1 & vamessages.log.2.

SCM (CVS, Subversion, and Perforce)

Software configuration management (SCM) servers generate several logs from the SourceForge-managed integration server in <SOURCEFORGE_INSTALL_DIR>/log. Not all of them are not unique to SourceForge; however, in the interest of completeness they are all documented here.

- catalina.out - This log contains information on the startup and runtime activities of the Tomcat server. This log is not rotated, nor is it overwritten, and is appended continuously over the lifetime of the server.
- localhost_log - This log contains a record of CVS or Subversion browsing URL construction. When a user attempts to browse a CVS or Subversion repository in his or her web browser, the URL construction process is documented in this log. This log is rotated for each date that there is activity.
Server logs

- localhost_admin_log - This log contains a record of the initial startup and deployment of the managed integration server. A new date stamped log is generated each time the integration server is started.
- vaexternalintegration.log - This log contains information on the operations that are being executed by the managed integration server. This log is stored in <SOURCEFORGE_INSTALL_DIR>/log.

Email

Both the SourceForge email and search backends are managed from a parent daemon known as Phoenix. The Phoenix daemon logs its activities to the phoenix.log which is stored under SOURCEFORGE_INSTALL_DIR/james/james-2.2.0/logs. This log is overwritten each time Phoenix is (re)started. Phoenix is run as part of the SourceForge standalone server init script. If the mail or search backends are not operating properly, the first troubleshooting step is to check the phoenix.log to see if either of them encountered difficulties starting up.

The SourceForge email has handled by the JAMES server. JAME logs all of its activities under SOURCEFORGE_INSTALL_DIR/james/james-2.2.0/apps/james/logs. Sixteen different logs are created by james for different components of james' functionality; however, most are not used actively by SourceForge, and are therefore not discussed here. A new log is created for each date when there is activity, and additional logs are created if james is restarted on a date when there is activity. The date is embedded in the log name (such as james-2005-04-28-01-00.log).

- james-$date.log - The James log records the overall mail handling behavior of the James server.
- mailet-$date.log - The mailet log records how each piece of email is handled. If there is a mail delivery problem, this log is the best place to begin investigation.
- mailstore-$date.log - The mailstore log records the behavior of mail spools, and the storage of mail. This log should normally not contain errors unless James is unable to write or read mail to or from the file system.
- smtpserver-$date.log - The smtpserver log records all inbound mail handling results. If email to discussion forums is not posting, or is getting rejected, this log would be the best place to begin investigation.
- spoolmanager-$date.log - The spoolmanager log records the processing of mail spools. This log could be of value in troubleshooting mail delivery or handling problems.

Search

Both the SourceForge email and search backends are managed from a parent daemon known as Phoenix. The Phoenix daemon logs its activities to the phoenix.log which is stored under SOURCEFORGE_INSTALL_DIR/james/james-2.2.0/logs. This log is overwritten each time Phoenix is (re)started. Phoenix is run as part of the SourceForge standalone server init script.
If the mail or search backends are not operating properly, the first troubleshooting step is to check the phoenix.log to see if either of them encountered difficulties starting up.

Once started successfully, the search server waits for new content needs to be indexed or searches to be performed. The search server logs its activities under SOURCEFORGE_INSTALL_DIR/james/james-2.2.0/apps/search/logs. The logs that are created are all named default with the date stamp appended to them (such as default-20041126.log). A new log is created for each date that there is indexing activity.

If the search server is not running, or expected search results are not being provided, the default log is the best place to investigate further.

**PostgreSQL**

PostgreSQL database logging is controlled and configured from two different places. The first is its init script, which is typically stored as /etc/init.d/postgresql. (See Appendix A for an example.) The second is in its configuration file, which is typically stored as /var/lib/pgsql/data/postgresql.conf. The postgresql init script includes a variable, PGLOG, which defines where the postgresql log is written. By default, PostgreSQL performs no logging at all until this variable is changed from its default, /dev/null, to a defined log file. /var/log/pgsql is a good choice; however, please note that this log will not be overwritten or rotated unless syslogd is configured to do so.

The postgresql.conf configuration file further defines what kind of information is then logged to the file specified by the PGLOG variable in the postgresql init script. The section titled 'ERROR REPORTING AND LOGGING' in postgresql.conf provides several parameters for configuring the content and verbosity of the log. Most of the defaults are usually fine for normal operations. However, it is recommended that the log_timestamp entry be uncommented and set to 'true' so that all log entries are accompanied by a date & time stamp, in case future troubleshooting is required.

Note that any changes to the postgresql init script or postgresql.conf will require that PostgreSQL be restarted before the changes take effect, and that SourceForge must be shut down prior to stopping (or restarting) PostgreSQL. Also note that PostgreSQL logging can degrade database performance, and thus should be configured and used carefully in a production environment.

For additional information, please see the official PostgreSQL documentation and consider joining the 'pgsql-general' mailing list:

http://www.postgresql.org/docs/8.2/interactive/runtime-config-logging.html

http://archives.postgresql.org/pgsql-general/

**Oracle**

An Oracle database performs logging on a wide array of functionality. The majority of the logs that are generated are stored under $ORACLE_HOME/admin/$SID/. While many different logs are stored under this directory hierarchy, the most important is the alert log.
Server logs

which is found in $ORACLE_HOME/admin/$SID/bdump/alert_$SID.log. This log records all
database activity, including serious problems. The alert log is not rotated or overwritten, and
can become quite large over time, especially on an active database.

Additional logs are created under the same directory hierarchy, for specific incidents. If a
problem is recorded in the alert log, the other logs should be inspected for additional details.

For additional information, as well as support in the maintenance of an Oracle database,
please contact Oracle Support and/or Oracle’s Metalink website:

http://metalink.oracle.com
SourceForge security

SourceForge® Enterprise Edition provides organizations with a secure, centralized, enterprise-grade solution for optimizing distributed development. It provides a web-accessible system for real-time project information.

J2EE Architecture

SourceForge Enterprise Edition 4.x is a J2EE application. It employs three-tier architecture to provide a secure environment for mission-critical data.

![SourceForge multi-tier architecture](image)

**Figure 1.** SourceForge multi-tier architecture

In a multi-tier architecture, access to each tier is restricted to the tier above it, thus securing the tiers behind the firewall effectively. For example, while clients (users accessing the system through a web) access the web server, they neither have access to the application and backend servers nor are they aware of their existence. Similarly, the web server itself does not have access to the backend servers (database, SCM, mail etc.)
Exceptions to this rule include:

- Direct client access provided to the SCM servers. SCM servers are accessed across the firewall typically through SSH protocol (for CVS), or HTTP or HTTPS (for Subversion). SCM server data is also accessible in a view only mode through the web interface.
- Clients must have access to the mail server for posting messages to mailing lists.
- Mail server must have access to deliver messages across the firewall.

Clients can also access the SOAP APIs through the web server. The web server in turn forwards SOAP requests to the application server for processing.

**Key elements of security**

The key elements of security behind the design of SourceForge include:

- Protection of sensitive data
- Effective data access control
- Activity tracking

**Protection of sensitive data**

Sensitive data must be protected from illegal access at various points in the system. Key areas where security is typically compromised include data transmission and data storage.

**Data transmission**

Network traffic is not encrypted by default. The HTTP protocol (non-SSL) does not protect data during transmission. HTTPS provides Strong Encryption using the Secure Socket Layer and Transport Layer Security protocols (SSL/TLS).

The web server employed by a SourceForge installation must be reconfigured to employ the HTTPS protocol.

**Data storage**

Sensitive data, such as credit card numbers, financial information, etc., must be stored securely. Usually this is done by encryption.

In the context of an application like SourceForge, sensitive data includes user passwords. SourceForge encrypts no other data. Since user passwords are used for authentication purpose only, SourceForge only stores password digests with an MD5 based cryptographic hash to guarantee adequate data protection.

MD5 is a one-way hash function that is used to verify data integrity through the creation of a 128-bit digest from data input. A one-way hash function is designed in such a way that it is hard to reverse the process, that is, to find a string that hashes to a given value. MD5 is
Currently, a standard, Internet Engineering Task Force (IETF) Request for Comments (RFC) 1321. According to the standard, it is “computationally infeasible” that any two messages that have been input to the MD5 algorithm could have as the output the same message digest, or that a false message could be created through apprehension of the message digest.

**Effective data access control**

In addition to data encryption, access to data must be strictly controlled to meet the security requirements of the enterprise. Strict data access control is achieved through a combination of firewalls, authentication, and authorization.

**Firewalls and network configuration**

A firewall provides the first level of protection by restricting access to the private network from the Internet. Sophisticated firewall configuration can provide strong security for all enterprise resources.

All SourceForge installations must be secured with a firewall restricting the access to specific web server ports. Neither the SourceForge application server nor the backend servers should ever be exposed to the Internet.

The SourceForge application and the backend servers can be further secured by limiting their access within the private network (Intranet). For SourceForge application to function effectively:

Across the firewall, clients (users) must have access to:

- The web server through a secure protocol such as HTTPS (port 443). The web server typically handles both the browser requests as well as the SOAP requests and forwards them to the SourceForge application server.
- Send mail to SourceForge mail server via SMTP (port 25).
- The SCM server through a secure protocol such as SSH (port 22).

In addition:

- The web server must have access to the application server (typically port 8080). Note that this port is not exposed outside the firewall.
- The web server must have access to the SCM server for repository browsing functionality.
- The application server must have access to the backend (SCM, database, mail, etc.) servers.
- The SCM server must be able to access SourceForge for commit notifications.
- The mail server must be able to deliver messages across the firewall.
Authentication and authorization

To secure sensitive data, SourceForge provides access control tools to restrict unauthenticated and non-member access.

User authentication is supported through verification of username and password during login. Project administrators can completely restrict access to authenticated members by marking projects as gated communities or private. A gated community is only accessible to unrestricted users, while a private project is only accessible to its members.

SourceForge provides fine grained access control through RBAC (Role-based access control). Users can be restricted to accessing specific application tools, object groups (trackers, task groups, etc) or they may be restricted to specific operations (such as the ability to view tracker artifacts but not create or update them). Project administrators can manage user access to projects using the administration tools provided by SourceForge.

Activity tracking

In case of a data security compromise, a record of who is performing what activities will help resolve some of the security issues.

Typically web servers log every page (or URL) being accessed, including the IP address of the user, date and time of access, etc. These logs are very useful in tracking the source of any security violations that may occur.

SourceForge application also audits every change made to application objects (trackers, artifacts, documents, etc.) within the system. Administrative actions such as adding a new user, assigning permissions to members are also tracked. Every change is recorded with the exact changes made to specific properties of objects within the system, the user making the change, the date and time the change was made, etc. SourceForge auditing tools are a powerful way to track unwanted and/or unauthorized changes within the system.

CERT advisories

CollabNet Product Support monitors the CERT coordination center (http://www.cert.org/) for notification of vulnerabilities or exploits against applications that SourceForge provides. If CollabNet Technical Support identifies an advisory that may indicate potential challenges for users who have deployed SourceForge, Support proactively releases a notification and a statement of action. CollabNet will provide product updates as it deems appropriate or necessary. Technical aspects of SourceForge security

This section discusses several technical aspects of SourceForge Enterprise Edition 4.4 security.
Cookies
SourceForge Enterprise Edition requires browsers to support cookies. Cookies are used for the sole purpose of managing user sessions. SourceForge uses session cookies for storing session ID information.

A transient cookie, sometimes called a session cookie, contains information about a user that disappears when the user’s browser is closed. Unlike a persistent cookie, a transient cookie is not stored on your hard drive but is only stored in temporary memory that is erased when the browser is closed.

Session management
SourceForge runs on the JBoss Application Server, with TomCat as the JSP/Servlet engine.
The JSP/Servlet engine is used for serving dynamic web pages and managing HTTP sessions. Servlet engines generate session IDs that are exchanged with the client browser as session (or transient) cookies.

TomCat generates Session IDs using the java.security.SecureRandom class. The java documentation for this class mentions the following:

This class provides a cryptographically strong pseudo-random number generator (PRNG). A cryptographically strong pseudo-random number minimally complies with the statistical random number generator tests specified in FIPS 140-2, Security Requirements for Cryptographic Modules, section 4.9.1. Additionally, SecureRandom must produce non-deterministic output and therefore it is required that the seed material be unpredictable and that output of SecureRandom be cryptographically strong sequences as described in RFC 1750: Randomness Recommendations for Security.

A user session is established after SourceForge authenticates a user’s login information. A session is invalidated when one of following events occur:

- The user explicitly logs out of SourceForge.
- When the user’s session times out.

Dismissing the browser leaves the session unusable until it is eventually timed out and invalidated.

Passwords
SourceForge only stores password digests with an MD5-based cryptographic hash to guarantee adequate data protection. MD5 is a one-way hash function. A one-way hash function is designed in such a way that it is hard to reverse the process, that is, to find a string that hashes to a given value.
Administrators can force SourceForge to reject passwords that do not meet a minimum password length. This feature is useful to help stop people from using trivial passwords where security is an issue.

By default, SourceForge does not perform any kind of password strength checking (e.g. SourceForge does not identify and reject dictionary-words/common names, does not expire passwords, does not enforce upper/lower case/special character combinations.) CollabNet’s Professional Services organization can install add-ons which enforce password expiration and other policies.

**Cross-site scripting (XSS) protection**

SourceForge Enterprise Edition 4.4 is designed to protect the application against Cross-site scripting (XSS) attacks. User-supplied text is encoded by clearing HTML markup before rendering it. Constant code reviews are performed to ensure that all fields are secured appropriately. High priority is given to fixing any oversights and issuing security patches as necessary.

**Ports**

The following table lists the ports that must be accessible across the firewall for SourceForge application to function properly:

<table>
<thead>
<tr>
<th>Server</th>
<th>Port</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web server</td>
<td>443 (SSL)</td>
<td>Users access the application web interface over SSL.</td>
</tr>
<tr>
<td>SCM server</td>
<td>22 (SSH)</td>
<td>Users typically access the SCM server over Secure Shell for performing checkins and checkouts.</td>
</tr>
<tr>
<td>Mail server</td>
<td>25 (SMTP)</td>
<td>Users send mail to post messages to the mailing lists (discussion forums).</td>
</tr>
</tbody>
</table>

**Summary**

SourceForge Enterprise Edition 4.4 provides a secure environment for distributed, enterprise development. In addition to employing industry standard security protocols, SourceForge Enterprise Edition provides an extensive access control model for fine-grain control and powerful tools to audit and track changes.

**Note** - Although CollabNet intends SourceForge Enterprise Edition as a secure, commercial application as delivered, it is not verified for highly secure computing environments that exceed an industry standard level of business application security. SourceForge Enterprise Edition may be extended to meet the specific needs of military, government or other highly secure facilities. Please contact CollabNet Professional Services if you have this requirement.
Firewalling, operating system, and Apache security

The various components of a SourceForge installation listen on a number of operating system ports; however, most are only used internally on the server for application communications. Thus, a very small subset must be exposed externally. This section will deal only with those parts that must be exposed externally, with the assumption that all others can be firewallled off to maintain security.

Following is a brief discussion of each operating system level port that must be exposed in order for users to access all SourceForge services:

- **22 (ssh)** - Port 22 is the default port for the secure shell (ssh). This is required for basic ssh functionality and for CVS, as all CVS transactions occur over ssh.
- **25 (smtp)** - Port 25 is the default port smtp (email). SourceForge discussion forums include mailing list functionality that allows users to send email to the SourceForge server. The James mail server included with SourceForge listens on port 25 to accept this mail for processing.
- **443 (https)** - Port 443 is the default port for encrypted web data transfer (https). The apache web server should be configured to encrypt all data so that it cannot be compromised by a 3rd party with malicious intent. Apache can be configured to force all traffic to be sent over https, even when a request is sent via port 80 (http).

To force all SourceForge traffic to use SSL encryption (HTTPS), you must edit your httpd.conf file and add a new mod_rewrite rule.

1. First save a copy of your current config file.
   ```
   cd /etc/httpd/conf
   install httpd.conf httpd.conf.pre_forced_ssl
   ```

2. Edit the file and add new rules:
   ```
   vi httpd.conf
   Find the line that reads:
   RewriteEngine on
   Modify the remaining lines to look like this:
   # Turn on mod_rewrite and define its logfile and log level
   RewriteEngine on
   RewriteLog logs/rewrite
   RewriteLogLevel 1
   # If the user came in over HTTP, redirect him or her to HTTPS and if they requested ‘/’ also redirect to ‘/sf’
   RewriteCond %{SERVER_PORT} ^80$
   RewriteRule ^/$ http://%{SERVER_NAME}/sf/ [R]
   ```
RewriteRule ^/(.*)$ https://%{SERVER_NAME}/$1 [R,L]
# If the user came in on '/sf' or '/sf/*' then pass it off to the app server
RewriteRule ^/sf$ http://localhost:8080/sf [P]
RewriteRule ^/sf/(.*) http://localhost:8080/sf/$1 [P]

# Pass ScmListener SOAP requests
RewriteCond %{REQUEST_URI} ^/sf-soap/services/ScmListener
RewriteRule ^/sf-soap/(.*) http://localhost:8080/sf-soap/$1 [P]

# Pass all non-listeners SOAP requests. Delete next 4 lines if you don’t use SOAP APIs.
RewriteCond %{REQUEST_URI} !^/sf-soap/services/[^/]*/Listener
RewriteRule ^/sf-soap/(.*) http://localhost:8080/sf-soap/$1 [P]
RewriteRule ^/sf-soap42/(.*) http://localhost:8080/sf-soap42/$1 [P]
RewriteRule ^/sf-soap43/(.*) http://localhost:8080/sf-soap43/$1 [P]

# User wants to use the SCM integration, so pass him or her off to that server

ProxyPassReverse / http://localhost:8080/
ProxyPassReverse / http://localhost:7080/

3. Save a copy of your current SSL config file.
   cd /etc/httpd/conf.d
   install ssl.conf ssl.conf.pre_forced_ssl

4. Edit the file, and add new rules:
   vi ssl.conf
   Find the line that reads:
   RewriteEngine on
   Modify the remaining lines to look like this:
   RewriteEngine on
   RewriteLog logs/rewrite_ssl
   RewriteCond %{SERVER_PORT} ^443$
   RewriteRule ^/$ https://%{SERVER_NAME}/sf/ [R,L]

   # If the user came in on '/sf' or '/sf/*' then pass it off to the app server
   RewriteRule ^/sf$ http://localhost:8080/sf [P]
   RewriteRule ^/sf/(.*) http://localhost:8080/sf/$1 [P]
# Pass ScmListener SOAP requests
RewriteCond %{REQUEST_URI} ^/sf-soap/services/ScmListener
RewriteRule ^/sf-soap/(.*) http://localhost:8080/sf-soap/$1 [P]

# Pass all non-listeners SOAP requests. Delete next 4 lines if you don’t use SOAP APIs.
RewriteCond %{REQUEST_URI} !^/sf-soap/services/[^/]*Listener
RewriteRule ^/sf-soap/(.*) http://localhost:8080/sf-soap/$1 [P]
RewriteRule ^/sf-soap42/(.*) http://localhost:8080/sf-soap42/$1 [P]
RewriteRule ^/sf-soap43/(.*) http://localhost:8080/sf-soap43/$1 [P]

# User wants to use the SCM integration, so pass him or her off to that server

ProxyPassReverse / http://localhost:8080/
ProxyPassReverse / http://localhost:7080/

5. Restart Apache.
   /etc/init.d/httpd restart # Red Hat Linux
   /etc/init.d/apache2 restart # SuSE Linux

6. Point your browser at SourceForge.
   It should automatically redirect to HTTPS traffic.

Please see Appendix D for a usable /etc/httpd/conf/httpd.conf and Appendix E for a usable /etc/httpd/conf.d/ssl.conf that has all of the above configuration changes implemented for a configuration where both the SourceForge application server and the only SCM (CVS, Subversion, or Perforce) integration server reside on the same physical server.
Generating Apache SSL certificates

Please note that if you wish to use and/or force all web traffic as https, you will need to obtain a valid apache SSL certificate. When generating an apache (mod_ssl) SSL certificate, you have two options:

- The first option is to purchase a SSL certificate from what is known as a 'certificate authority' (CA). There are many CAs available, and CollabNet does not recommend one over another. Going to Google and searching for "certificate authority" will present several choices.

- The second is to generate what is known as a 'self signed' certificate. This option costs nothing and provides the same level of encryption as a certificate purchased from a certificate authority (CA), however some web browsers, primarily Internet Explorer (IE), will issue a harmless warning each time a user visits a site that uses a self-signed certificate. Thus, this option can be a mild annoyance to some users.

Regardless of which option you select, the process is almost identical:

1. First you have to know the Fully Qualified Domain Name (FQDN) of the website for which you want to request a certificate. When you want to access your website through https://www.example.com then the FQDN of your website is www.example.com. www.example.com will be your common name.

2. Generate the key with the following command:

   ```bash
   openssl genrsa -out www.example.com.key 1024
   ```

   This command generates 1024 bit RSA Private Key and stores it in the file www.example.com.key. Backup your www.example.com.key file, as without this file your SSL certificate will not be valid.

3. Generate the CSR with the following command:

   ```bash
   openssl req -new -key www.example.com.key -out www.example.com.csr
   ```

   This command will prompt you for the X.509 attributes of your certificate. Remember to give the name www.example.com when prompted for `Common Name (eg, YOUR name)'. Do not enter your personal name here. It is requesting a certificate for a webserver, so the Common Name has to match the FQDN of your website (a requirement of web browsers).

4. Generate a self-signed Certificate with the following command:

   ```bash
   ```

   This command will generate a certificate a self-signed certificate in www.example.com.crt which can be used as a temporary (or permanent) certificate if you are purchasing a real certificate from a CA.
You will now have a RSA Private Key in www.example.com.key, a Certificate Signing Request in www.example.com.csr, and an SSL certificate in www.example.com.crt. The self-signed SSL certificate that you generated will be valid for 370 days.

**Further securing HTTPS**

Deploying an Apache SSL certificate and forcing https ensures that all data is encrypted. It does not, however, ensure that the encryption methods (also known as ciphers) that are used are strong. With the ever-increasing power of today's computers, many older or weaker ciphers can be cracked in a matter of days or even hours by a determined person with malicious intentions.

In order to mitigate this risk, it is advisable to only allow the strongest ciphers available. Edit /etc/httpd/conf.d/ssl.conf and look for either SSLProtocol or SSLCipherSuite. If these exist, then you will need to alter them. If they do not exist, you will need to add them. Simply add them below the SSLEngine line.

You will need to add the following two lines:

```
SSLProtocol all -SSLv2
SSLCipherSuite RSA:!EXP:!NULL:+HIGH:+MEDIUM:-LOW
```

Save the file, and restart Apache (apachectl restart).

Please see Appendix D for a usable /etc/httpd/conf/httpd.conf and Appendix E for a usable /etc/httpd/conf.d/ssl.conf that has all of the above configuration changes implemented.
Preventing HTTP Trace/Put/Delete methods

Apache includes several methods that can be used to manipulate data. These include the trace, put, and delete methods. These have certain uses, but are not used by SourceForge and can be abused by those with malicious intentions. Therefore it is recommended that you disable all three methods to prevent their exploitation unless you have a need to use them.

All of the following instructions require that you edit /etc/httpd/conf/httpd.conf.

To fix the trace issue, search for the RewriteEngine line and make sure it is set to 'on'.

Then add the following two lines:

```bash
RewriteCond %{REQUEST_METHOD} ^TRACE
RewriteRule .* - [F]
```

To fix the put and delete issues, search for the following line:

```bash
<Directory "/var/www/html">
```

Once you have found this line, you will need to scroll down in the file to find the Order line.

Now replace the Order and Allow lines with:

```bash
<Limit GET POST OPTIONS PROPFIND>
  Order allow,deny
  Allow from all
</Limit>

<Limit PUT DELETE PATCH PROPPATCH MKCOL COPY MOVE LOCK UNLOCK>
  Order deny,allow
  Deny from all
</Limit>
```

Now save the file, and restart Apache:

```bash
apachectl restart
```

Please see Appendix D for a usable /etc/httpd/conf/httpd.conf and Appendix E for a usable /etc/httpd/conf.d/ssl.conf that has all of the above configuration changes implemented.

Additional information on these methods and the risks involved with their activation can be found at the following links:

- [http://www.apacheweek.com/issues/03-01-24#news](http://www.apacheweek.com/issues/03-01-24#news)
- [http://www.apacheweek.com/features/put](http://www.apacheweek.com/features/put)
**Recommended firewall configuration**

Based on the discussion in the previous section, only ports 22, 25 and 443 must be exposed on a SourceForge server as long as you are running the only SCM (CVS, Subversion, or Perforce) integration server on the same server as the SourceForge standalone application. See Appendix C for RedHat Linux iptables rules that will enforce the security configuration noted above. Appendix C can be used in its current state on the SourceForge server as /etc/sysconfig/iptables.

However, if you are running the SCM (CVS, Subversion, or Perforce) integration server on a separate physical server from the SourceForge standalone application server, then you must expose port 7080 on the SCM integration server so that the application server can communicate with the SCM integration server.

**Password security**

SourceForge includes functionality to control the minimum length of the passwords that users select. By default, SourceForge requires passwords that are at least six (6) characters long, however you can increase this minimum length by adding the following variable

```
  system.password.min-length=8
```

to

```
  SOURCEFORGE_INSTALL_DIR/sourceforge_home/etc/sourceforge_configuration.properties
```

By adding the above variable, you will set a new minimum password length of eight (8) characters. SourceForge will need to be restarted for this change to take effect, and will only impact new user registration and users who change their passwords in SourceForge.

Third party application support limitations

SourceForge relies on many third party applications to augment or enhance functionality. However, there are known limitations to both support and functionality in some of these applications. Known limitations are discussed in this section.

CVS

CollabNet SourceForge Technical Support provides 'best effort' support for CVS client usage issues. As SourceForge does not ship any CVS functionality (neither the server daemon nor the client ships with SourceForge,) Support makes a best effort to assist with CVS usage questions as well as in the event that they are due to a problem with SourceForge functionality. Ultimately, customers should contact the vendor that supplied the CVS server and client(s) that they are using for assistance.

The CVS RPM that ships with RedHat Linux Enterprise Server 3 and RedHat Advanced Server 2.1 has a known bug that prevents users who have access to thirty-two (32) or more CVS repositories from accessing the repositories that are alphabetically after the thirty-first (31st). This is currently RedHat bug #131124.

([https://bugzilla.redhat.com/bugzilla/show_bug.cgi?id=131124](https://bugzilla.redhat.com/bugzilla/show_bug.cgi?id=131124))

Customers are advised to contact RedHat for a solution to this bug if they have SourceForge users who are a member of thirty-two (32) or more CVS repositories on a SourceForge CVS server.

Discussion forum threading

In order for SourceForge discussion forums to properly thread posts sent in via email, the email message must include either the References or In-Reply-To header. Email received without both of those headers cannot be threaded accurately and will most likely be treated as a new topic or thread in the discussion. The In-Reply-To and References headers are strongly recommended to be included in the header for the reply to any email message in section 3.6.4 of RFC-2822. While the lack of either of the headers is not an explicit RFC violation, the inclusion of such headers is considered RFC compliance. RFC-2822 is available for review here:


Unfortunately, two of the more popular email clients, Microsoft Outlook and Lotus Notes, are prone to not sending mail with at least one of the required headers. There is evidence that Lotus Notes versions 6.5 and newer are capable of sending email that includes at least one of the two required headers. However older versions of Lotus Notes either do not include the headers, or require special reconfiguration in order to do so.
Microsoft Outlook on its own does include the In-Reply-To header. However, when used to send mail through a Microsoft Exchange server, that header is stripped off. There are no known versions of Microsoft Exchange server that do not strip these RFC headers from outbound email, and as such there are no known workarounds at this time.

Customers are advised to contact their IT group and/or the vendor of their email client with questions or concerns.

Support for other third party applications

SourceForge integrates with additional third party applications, such as Microsoft Office 2003 and XP, and Microsoft Project 2002 & 2003. Support will always make an effort to provide assistance in using third party applications with SourceForge. However, for complete, end-to-end support, customers should consult with the application vendor, as the vendor is always best equipped to provide the depth and breadth of support necessary to use their products.
SourceForge and operating system maintenance

This section discusses various suggestions for maintaining your SourceForge installation.

SourceForge Hot Fixes and Service Packs

CollabNet releases periodic updates for each version of SourceForge that is currently supported, in the form of Hot Fixes and Service Packs.

Hot Fixes

A SourceForge Hot Fix is released to resolve one or more high priority bugs (typically priority one) that require an urgent customer solution. A README is included with each Hot Fix. The README documents the bug(s) that are resolved and provides installation instructions.

Service Packs

A SourceForge Service Pack (SP) is an accumulation of many bug fixes (including all Hot Fixes since the last Service Pack) as well as select high-priority customer-requested enhancements. Both a README and Release Notes are included with each Service Pack. The README documents the installation procedure and lists the SourceForge files that the Service Pack changes, adds, or removes. The Release Notes documents all of the bugs and enhancements that the Service Pack addresses, as well as any other application or documentation changes. Service Packs must be applied in the order in which they were released (numerical order).
Operating system maintenance

SourceForge is currently tested and qualified to run on Red Hat Advanced Server 3, Red Hat Enterprise Server 3, Red Hat Enterprise Server 4, and Novell SUSE Linux Enterprise 9 as well as CentOS 4.

Red Hat provides package updates for both operating system releases in order to address security and stability issues as they arise. Customers are encouraged to subscribe to Red Hat’s enterprise-watch mailing list to be notified when updates are released. To subscribe to the mailing list, please point your web browser to the following URL:

https://www.redhat.com/mailman/listinfo/enterprise-watch-list

For SuSE Linux Enterprise updates and notices, subscribe via the URL:

http://support.novell.com/suseNotifications/index.jsp

For CentOS updates and notices, subscribe via the URL:

http://lists.centos.org/mailman/listinfo/centos-announce

SourceForge Support resources

The SourceForge Support team provides several services to customers to help ensure a smooth and positive experience when using SourceForge. This section elaborates on these offerings.

Submitting feature requests

Please provide a detailed description of the functionality or enhancement that you would like added to the product or describe how you’d like the product to act differently. The more specific you are, the better we will be able to meet your needs. Please be advised that CollabNet may generalize your request to make it satisfy similar needs from other customers.

Additionally, please provide the business driver or use case for your request. The better that you can describe how this feature would be utilized and how important it is to your organization, the better equipped we will be to assign the correct priority to your request.

Also, let us know which version of SourceForge you are currently using. This information can be found in the 'About SourceForge' screen in the 'Help' section of the product.

Lastly, please justify the priority that you feel that this new functionality or enhancement has in your organization. Please use the following priorities as a guide:

- Priority 1 - This is ‘must have’ functionality. Without this functionality there is significant business or commercial impact to your organization or user base.
Priority 2 - This is strongly needed functionality. The need is not urgent at this time, but it may be urgent at some time in the future.

Priority 3 - This is a 'nice to have' feature or enhancement. Obtaining this new functionality would improve the user experience in some concrete way, but there is little or no impact to your organization or user base without this functionality.

Priority 4 - Seems useful, and/or the demand is low within your organization.

Combine all this information into an email to Support, and Support will submit your request to the Product Management group.

Although CollabNet cannot commit to the inclusion of a particular feature request in a specific release, all feature requests submitted are reviewed during the planning phases of each new major SourceForge release.

**Submitting a bug report**

Before contacting SourceForge Support to file a bug report you may wish to do the following:

- Review the SourceForge KnowledgeBase (http://sfee.open.collab.net/sf/wiki/do/viewPage/projects.kb/wiki/HomePage) for information related to the question or problem you are experiencing. The SourceForge KnowledgeBase contains a wealth of information about SourceForge installation, configuration and application use. For more information on accessing the KnowledgeBase, see “The SourceForge KnowledgeBase” on page 65.

- Review the Release Notes from recent Service Packs. You may find that your issue or feature request has been resolved or included in a recent Service Pack, or is already listed as a 'Known Issue'. The Release Notes are available for download with each Service Pack on the SourceForge download project (https://sfee.open.collab.net/sf/projects/sfdl). For more information on accessing the SourceForge download project, see “The SourceForge Download Project” on page 66.

- If neither the KnowledgeBase nor the Service Pack Release Notes address your issue, please collect the following information and have it ready for inclusion in your email to Support:

  First, you should be able to fully describe what the application did that you feel is incorrect. You should include a description of the steps needed to trigger the behavior, and what the expected behavior should have been.

  Next, you need to be able to define the scope of the problem.

  1. Does it affect:
     - Just you?
     - Multiple users?
     - All users?
2. Does the problem occur in:
   • One project/tracker/artifact/forum/task/etc.?
   • Some projects/tracker/artifact/forum/task/etc.?
   • Projects/tracker/artifact/forum/task/etc migrated from a 3.x installation?
   • All projects/tracker/artifact/forum/task/etc.?

3. Is the account you used:
   • A SourceForge 'admin' account?
   • A normal account

Additionally, you will need to provide the following information:

- Identify the version(s) of software involved.
- What version of SourceForge are you running? This is found on the 'About SourceForge' screen in the 'Help' section of the application.
- Which web browser and version are you using? You can find this by going to the 'About' screen of the browser's 'Help' menu.

Finally, please justify the priority that you feel that this bug or defect has to your organization or users with the following priorities as a guide:

- Priority 1 - Severe impact, SourceForge-wide loss of functionality, data loss, or immediate impact to primary business plans or user base, with no workaround or avoidance mechanism. A solution is needed urgently, and cannot be deferred to a future major SourceForge release.
- Priority 2 - Significant impact, isolated, or limited loss of functionality, impact to a subset of users, and a workaround or avoidance mechanism is available. A solution is desired in the next major SourceForge release or Service Pack.
- Priority 3 - Most bugs will fall into this category. Impact is low or minor, to the extent that either a workaround exists, or the loss of functionality has no significant impact on daily SourceForge usage by the majority of users. A solution is desired in the next major SourceForge release.
- Priority 4 - Little or no impact or loss of functionality to the majority of users. A solution is preferred, but is not required in the next major SourceForge release.

Now that you have all the relevant information, place it into an email to Support and we'll be happy to examine the bug and have it prioritized.

The SourceForge KnowledgeBase

Support maintains a KnowledgeBase website that is regularly updated to include individual articles that address the questions and issues that customers most frequently ask Support. The articles are broken down into categories to aid in quickly locating information. The
SourceForge Support resources

SourceForge KnowledgeBase is open to all customers, without restrictions, by pointing your web browser to:


Comments, questions, and feedback on the SourceForge KnowledgeBase are always appreciated by sending email to Support at support-sf@vasoftware.com.

The SourceForge Download Project

All supported SourceForge software, documentation and updates are hosted on the SourceForge download project:

https://sfee.open.collab.net/sf/projects/sfdl

Access is restricted to named primary and secondary customer contacts listed on the original SourceForge purchase agreement. Access can be granted to additional named individuals by making a request to Support from anyone who currently has access.
The default configuration as shipped with PostgreSQL is adequate for most SFEE installs with the following modifications to the /var/lib/pgsql/data/postgresql.conf:

listen_address = <database_host_ip_address specified in install-saturn.conf>
shared_buffers = 50000
effective_cache_size = 65536
geo = false

Please note that you will need to restart PostgreSQL for these changes to take affect

PostgreSQL 8 adds support for automated vacuuming of the database, which runs as necessary based on actual activity statistics. The following lines are one way to configure "Auto Vacuum".

stats_start_collector = true
stats_row_level = true
superuser_reserved_connections = < old value plus 1> (one slot is needed for the autovacuum process)
autovacuum_naptime = <number of seconds to run the autovacuum. Large sites = 300. Small sites = 14400>
This appendix provides /etc/sysconfig/iptables that will enforce the security configuration recommended in “Firewalling, operating system, and Apache security” on page 53. For additional information on recommended firewall configuration, see “Recommended firewall configuration” on page 59.

```
# 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 25 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
-A RH-Firewall-1-INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A RH-Firewall-1-INPUT -j REJECT --reject-with icmp-host-prohibited
COMMIT
```
These SFEE-specific changes need to be made to the /etc/httpd/conf/httpd.conf file.

## SFEE configuration

##

# mod_deflate for improving performance
DeflateFilterNote Input instream
DeflateFilterNote Output outstream
DeflateFilterNote Ratio ratio
LogFormat '"%r" %{outstream}n/%{instream}n %{ratio}n%%' deflate

<Location />
  AddOutputFilterByType DEFLATE text/html
  # Netscape 4.x has some problems...
  BrowserMatch ^Mozilla/4 gzip-only-text/html
  # Netscape 4.06-4.08 have some more problems
  BrowserMatch ^Mozilla/4\.0\[678\] no-gzip
  # NOTE: Due to a bug in mod_setenvif up to Apache 2.0.48
  # the above regex won't work. You can use the following
  # workaround to get the desired effect:
  BrowserMatch \bMSIE\b no-gzip
  # Don't compress images
  SetEnvIfNoCase Request_URI .\.(?:gif|jpe?g|png)$ no-gzip dont-vary
  # Make sure proxies don't deliver the wrong content
  Header append Vary User-Agent env=!dont-vary
</Location>

# mod_expires for even better performance
ExpiresActive On
ExpiresDefault "access plus 0 seconds"
ExpiresByType image/gif "access plus 1 days"
ExpiresByType image/jpeg "access plus 1 days"
ExpiresByType image/png "access plus 1 days"
ExpiresByType text/css "access plus 7 days"
ExpiresByType text/javascript "access plus 7 days"
ExpiresByType application/x-javascript "access plus 7 days"
ExpiresByType image/x-icon "access plus 7 days"

# SFEE rewrites to make the app 'live' on port 80 and not 8080
RewriteEngine on
RewriteLog logs/rewrite
RewriteLogLevel 1
# Added to supress http trace for security reasons
RewriteCond %{REQUEST_METHOD} ^TRACE
RewriteRule .* - [F]
# make '/' redirect to SFEE
RewriteRule ^/$ http://%{SERVER_NAME}/sf/ [R]
# now pass the URL to the actual SFEE application server
RewriteRule ^/sf$ http://localhost:8080/sf [P]
RewriteRule ^/sf/(.*) http://localhost:8080/sf/$1 [P]

# Pass ScmListener SOAP requests
RewriteCond %{REQUEST_URI} ^/sf-soap/services/ScmListener
RewriteRule ^/sf-soap/(.*) http://localhost:8080/sf-soap/$1 [P]
# Pass all non-listeners SOAP requests. Delete next 4 lines if you don’t use SOAP APIs.
RewriteCond %{REQUEST_URI} !^/sf-soap/services/[^/] Listener
RewriteRule ^/sf-soap/(.*) http://localhost:8080/sf-soap/$1 [P]
RewriteRule ^/sf-soap42/(.*) http://localhost:8080/sf-soap42/$1 [P]
RewriteRule ^/sf-soap43/(.*) http://localhost:8080/sf-soap43/$1 [P]

# route SCM requests to the SFEE integration server
RewriteCond %{REQUEST_URI} !^/integration/services
RewriteCond %{REQUEST_URI} !^/integration/servlet
ProxyPassReverse / http://localhost:8080/
ProxyPassReverse / http://localhost:7080/

##
# end SFEE configuration
##
These are the SFEE-specific changes that need to be made to the /etc/httpd/conf/ssl.conf file.

```
##
# SFEE configuration
##
# mod_deflate for improving performance
DeflateFilterNote Input instream
DeflateFilterNote Output outstream
DeflateFilterNote Ratio ratio
LogFormat "\"%r\" \%{outstream}n/%{instream}n %{ratio}n\%\"' deflate
<Location />
  AddOutputFilterByType DEFLATE text/html
  # Netscape 4.x has some problems...
  BrowserMatch ^Mozilla/4 gzip-only-text/html
  # Netscape 4.06-4.08 have some more problems
  BrowserMatch ^Mozilla/4\.0[678] no-gzip
  # NOTE: Due to a bug in mod_setenvif up to Apache 2.0.48
  # the above regex won't work. You can use the following
  # workaround to get the desired effect:
  BrowserMatch \bMSI\[E\] no-gzip
  # Don't compress images
  SetEnvIfNoCase Request_URI \n  \.(?::gif|jpe?g|png)$ no-gzip dont-vary
  # Make sure proxies don't deliver the wrong content
  Header append Vary User-Agent env=!dont-vary
</Location>

# mod_expires for even better performance
ExpiresActive On
```
ExpiresDefault "access plus 0 seconds"
ExpiresByType image/gif "access plus 1 days"
ExpiresByType image/jpeg "access plus 1 days"
ExpiresByType image/png "access plus 1 days"
ExpiresByType text/css "access plus 7 days"
ExpiresByType text/javascript "access plus 7 days"
ExpiresByType application/x-javascript "access plus 7 days"
ExpiresByType image/x-icon "access plus 7 days"

# SFEE rewrites to make the app 'live' on port 80 and not 8080
RewriteEngine on
RewriteLog logs/rewrite
RewriteLogLevel 1
# Added to supress http trace for security reasons
RewriteCond %{REQUEST_METHOD} ^TRACE
RewriteRule .* - [F]
# make '/' redirect to SFEE
RewriteRule ^/$ https://%{SERVER_NAME}/sf/ [R]

# Pass ScmListener SOAP requests
RewriteCond %{REQUEST_URI} ^/sf-soap/services/ScmListener
RewriteRule ^/sf-soap/(.*) http://localhost:8080/sf-soap/$1 [P]

#Pass all non-listeners SOAP requests. Delete next 4 lines if you don’t use SOAP APIs.
RewriteCond %{REQUEST_URI} !^/sf-soap/services/[^/]*Listener
RewriteRule ^/sf$ http://localhost:8080/sf [P]
RewriteRule ^/sf/(.*) http://localhost:8080/sf/$1 [P]
# expose the SOAP API
RewriteRule ^/sf-soap/(.*) http://%{SERVER_NAME}:8080/sf-soap/$1 [P]
RewriteRule ^/sf-soap42/(.*) http://%{SERVER_NAME}:8080/sf-soap42/$1 [P]
RewriteRule ^/sf-soap43/(.*) http://%{SERVER_NAME}:8080/sf-soap43/$1 [P]
# route SCM requests to the SFEE integration server
RewriteCond %{REQUEST_URI} !^/integration/services
RewriteCond %{REQUEST_URI} !^/integration/servlet
ProxyPassReverse / http://localhost:8080/
ProxyPassReverse / http://localhost:7080/

##
# end SFEE configuration
##

Index

A
Apache
  serving static files 11
  SourceForge configuration 32

C
configuration file, SourceForge
  editing 24
  internationalization parameters 27
  parameters 24
configuration, Apache
  SourceForge 32
CVS
  server setup 14

D
database preparation
  Oracle 9
  PostgreSQL 8

H
hardware requirements 2

I
installation, SCM
  finishing 33
installation, SourceForge
  Apache configuration 32
  database preparation, Oracle 9
  database preparation, PostgreSQL 8

failure 30
finishing 32
installation program 30
options 5
overview 5
procedures 6
server preparation on Linux 11
internationalization 27

L
LDAP authentication 34

O
Oracle preparation 9

P
packages, installing 13
parameters, configuration file 24
Perforce
  server setup 22
PostgreSQL
  preparation for SourceForge 8

S
SCM Integration Server 13
SCM server
  adding 38
  configuration script 38
  finishing installation 33
  software requirements 2
SourceForge server installation
  Apache 32
  configuration file 24
  failure 30
  finishing 32
  installation program 30
  preparation on Linux 11
static files 11
Subversion
  server setup 20
system requirements 2

T
third party packages 13