

CollabNet TeamForge 5.2

System Administrator Guide

2009
CollabNet Inc.

Contents

Legal fine print.....	7
CollabNet, Inc. Trademark and Logos.....	7
Chapter 1: Set up a CollabNet TeamForge 5.2 site	9
Install a dedicated CollabNet TeamForge 5.2 site.....	10
Set up hardware for CollabNet TeamForge 5.2	10
Get CollabNet TeamForge 5.2	10
Configure your CollabNet TeamForge 5.2 site.....	11
Install the CollabNet TeamForge 5.2 application.....	12
Supply your TeamForge license key.....	12
Install an advanced CollabNet TeamForge 5.2 site	12
Set up hardware for an advanced CollabNet TeamForge 5.2 installation	13
Get CollabNet TeamForge 5.2	13
Configure your CollabNet TeamForge 5.2 site.....	14
Install the CollabNet TeamForge 5.2 application.....	15
Set up the database for your CollabNet TeamForge 5.2 site.....	16
Start your new TeamForge site.....	18
Supply your TeamForge license key.....	18
Install CollabNet TeamForge 5.2 without Internet access.....	19
Install CollabNet TeamForge 5.2 on a virtual machine.....	20
Get CollabNet TeamForge 5.2	20
Configure CollabNet TeamForge 5.2	21
Supply your TeamForge license key.....	22
Upgrade a CollabNet SourceForge Enterprise 5.1 site to CollabNet TeamForge 5.2	22
Uninstall CollabNet TeamForge 5.1.....	23
Get CollabNet TeamForge 5.2	23
Get the CollabNet TeamForge 5.2 update package.....	23
Install the CollabNet TeamForge 5.2 application.....	24
Finish your upgrade.....	25
Upgrade a CollabNet SourceForge Enterprise 5.1 site to CollabNet TeamForge 5.2 on a virtual machine.	26
Get the CollabNet TeamForge 5.2 upgrade package.....	26
Update CollabNet TeamForge 5.2	26
Start CollabNet TeamForge 5.2	27
Upgrade to TeamForge 5.2 without Internet access.....	27
Install a different build of the same release.....	28
Chapter 2: Support CollabNet TeamForge users.....	31
Authenticate users with LDAP.....	32
Set up LDAP integration for the CollabNet TeamForge 5.2 server.....	32
Set up LDAP for a source control integration server.....	33
Modify the application policy.....	36

Let users see what's in a project template.....	36
Chapter 3: Grow your CollabNet TeamForge installation.....	37
Set up Subversion on its own server.....	38
Set up the database for your CollabNet TeamForge 5.2 site on a separate server.....	38
Set up a PostgreSQL database on its own server.....	39
Set up an Oracle database on its own server.....	39
Chapter 4: Protect your CollabNet TeamForge site.....	41
Set up SELinux.....	42
Protect Apache with SSL.....	42
Set up Apache for SSL encryption.....	42
Generate Apache SSL certificates.....	43
Prevent HTTPS cracking.....	43
Protect integrations with SSL.....	44
Set up SSH tunneling.....	45
Chapter 5: Maintain your CollabNet TeamForge installation.....	47
Monitor services on your site.....	48
Get information about a CollabNet TeamForge 5.2 site.....	48
Patch CollabNet TeamForge 5.2	48
Go to an arbitrary patch level.....	49
Revert a patch upgrade or downgrade.....	49
Remove a patch.....	50
Troubleshoot patches.....	50
Specify DNS servers.....	51
Optimize PostgreSQL with vacuum.....	51
Change the location of a log file.....	52
Back up and restore CollabNet TeamForge 5.2 data.....	52
Back up CollabNet TeamForge 5.2 data.....	52
Restore backed-up CollabNet TeamForge 5.2 data.....	53
Back up a PostgreSQL database.....	53
Restore a PostgreSQL database.....	53
Appendix A: Frequently asked questions about CollabNet TeamForge system administration.	55
What does it take to install CollabNet TeamForge 5.2?.....	56
Do I need an advanced TeamForge 5.2 install?.....	56
How many servers do I need to run a CollabNet TeamForge 5.2 site?.....	56
Which application runs on which server?.....	57
How does CollabNet TeamForge 5.2 handle third-party applications?.....	57
Which ports should I keep open?.....	58
What does it mean to run CollabNet TeamForge 5.2 on a virtual machine?.....	59
Why won't my CollabNet TeamForge 5.2 virtual machine installation start?.....	59
Why does my CollabNet TeamForge 5.2 site show a different time than the host machine it is running on?.....	59

How does CollabNet TeamForge 5.2 manage security?.....	60
How does CollabNet TeamForge 5.2 help protect data access?.....	60
What user activities are tracked?.....	60
How does CollabNet TeamForge 5.2 help protect my data?.....	61
Does CollabNet TeamForge 5.2 work with LDAP?.....	61
J2EE Architecture and security.....	61
What security tools come with CollabNet TeamForge 5.2?.....	62
What is a CERT advisory?.....	63
What is a patch?.....	63
Does CollabNet TeamForge 5.2 support merge tracking?.....	64
Should I move my TeamForge database to its own server?.....	64
Should I move my source control application to its own server?.....	64

Appendix B: Reference information for CollabNet TeamForge system administration.65

Install reference.....	66
Minimum hardware requirements for CollabNet TeamForge 5.2	66
Supported software for CollabNet TeamForge 5.2	66
Packages required for 32-bit Red Hat 5.....	67
Packages required for 32-bit Red Hat 4.....	68
Packages required for 64-bit Red Hat 4.....	68
Hardware and software requirements for the CollabNet TeamForge 5.2 download.....	69
Contents of the CollabNet TeamForge 5.2 download package.....	69
Scripts.....	70
bootstrap-data.sh.....	70
The collabnet script.....	70
environment_check.sh.....	71
install.sh.....	72
pbl.py.....	73
snapshot.py.....	73
upgrade-site.sh.....	74
Log files.....	75
JBoss logs.....	75
Oracle logging.....	76
SCM (CVS, Subversion, and Perforce) logs.....	76
Email logs.....	76
Search logs.....	77
Project Build Library audit log	77
Profile audit log.....	78
User Audit Log.....	78
Host audit log.....	78
Project audit log.....	79
Configuration files.....	79
The patch manifest file.....	79
login-config.xml.....	80
httpd.conf.....	80
iptables.....	81

Legal fine print

Copyright © 2007 CollabNet, Inc. All rights reserved.

CollabNet is a trademark or registered trademark of CollabNet, Inc., in the U.S. and other countries. All other trademarks, brand names, or product names belong to their respective holders.

CollabNet makes no representation with respect to accuracy or completeness of document, and specifically disclaims any implied warranties for any purpose and shall in no event be liable for any loss of profit or any other commercial damage, including but not limited to special, incidental, consequential, or other damages.

Please submit comments and questions at www.collab.net.

CollabNet, Inc. Trademark and Logos

These trademarks are trademarks or registered trademarks of CollabNet, Inc. or its licensors in the United States and other countries.

- CollabNet
- OpenCollabNet
- CollabNet TeamForge
- CollabNet SourceForge Enterprise
- SourceForge Enterprise Edition
- Powered by CollabNet™
- collabXchange™

This list will be updated from time to time to reflect additional trademarks and changes in registration status. If you have questions or would like further information regarding CollabNet's trademarks, please contact the CollabNet, Inc., Legal Department at trademarks@collab.net.

Subversion is a registered trademark of the Subversion Corporation.

All other trademarks, logos, brand names, or product names belong to their respective holders.


Chapter 1

Set up a CollabNet TeamForge 5.2 site

Topics:


- [Install a dedicated CollabNet TeamForge 5.2 site](#)
- [Install an advanced CollabNet TeamForge 5.2 site](#)
- [Install CollabNet TeamForge 5.2 without Internet access](#)
- [Install CollabNet TeamForge 5.2 on a virtual machine](#)
- [Upgrade a CollabNet SourceForge Enterprise 5.1 site to CollabNet TeamForge 5.2](#)
- [Upgrade a CollabNet SourceForge Enterprise 5.1 site to CollabNet TeamForge 5.2 on a virtual machine](#)
- [Upgrade to TeamForge 5.2 without Internet access](#)
- [Install a different build of the same release](#)

You can set up CollabNet TeamForge 5.2 automatically or in a more complex custom configuration. For guidance on choosing between a dedicated install and an advanced install, see [Do I need an advanced TeamForge 5.2 install?](#) on page 56

 **Note:** If you are installing the trial version of CollabNet TeamForge 5.2 on VMware, use the instructions at [Install CollabNet TeamForge 5.2 on a virtual machine](#) on page 20 instead of these instructions.

Install a dedicated CollabNet TeamForge 5.2 site

A dedicated installation is the easiest way to set up CollabNet TeamForge. Minimal configuration is required. All services run on a single server, which the installer sets up.


 **Note:** Your server should run only CollabNet TeamForge. If you need to run other applications on the same system, or if other applications on your network must share the site's database or source control services, see [Install an advanced CollabNet TeamForge 5.2 site](#) on page 12.

Set up hardware for CollabNet TeamForge 5.2

To prepare the computer that your site will run on, install Linux and set up networking.


- For information about the operating systems you can use with CollabNet TeamForge 5.2, see [Supported software for CollabNet TeamForge 5.2](#) on page 66.
- For the hardware requirements, see [Minimum hardware requirements for CollabNet TeamForge 5.2](#) on page 66

You must have root-level access to the machine where you are installing CollabNet TeamForge 5.2

 **Important:** Only CollabNet TeamForge 5.2 should be using the TeamForge database and source control services. If any other application is dependent on the same database application or source control service, follow the configuration instructions at [Install an advanced CollabNet TeamForge 5.2 site](#) on page 12.

1. Install the operating system according to the instructions from the OS provider.
 - On Red Hat 5.3 or CentOS 5.2, do not customize your installation. Take the default settings only.
 - On Red Hat 4.7 or CentOS 4.7, select the Minimal installation option.

2. Make sure port 80 is open.


 **Tip:** For detailed firewall requirements, see [Which ports should I keep open?](#) on page 58

3. Disable SELinux.
4. Verify that the machine name is resolvable on the network.
 - a) Use the `hostname` command to verify the name of the machine.

```
hostname
bigbox.supervillain.org
```

- b) Use the `nslookup` command to verify that your hostname maps to the right IP address.

```
nslookup bigbox.supervillain.org
Server: 204.16.107.137
Address: 204.16.107.137#53
```

 **Tip:** If there is any doubt about what the system's real IP address is, use the `/sbin/ifconfig` command.

Your server is ready. You can now set up the software environment in which you will install CollabNet TeamForge 5.2.


Get CollabNet TeamForge 5.2

Download the packages you need to install and set up CollabNet TeamForge 5.2.

1. If you are installing on Red Hat Enterprise Linux 4.7, you must install the yum package manager.
 - a) Download the yum installer from <http://yum.collab.net/yum-install.sh>.
 - b) Run the yum installer.

```
sh yum-install.sh
```


2. Create the directory where you will save the TeamForge installer.

 **Tip:** You can put the installer anywhere, but for simplicity we recommend `/opt/collabnet/teamforge-installer`. All the examples given in these instructions use that path.

```
mkdir -p /opt/collabnet/teamforge-installer
```

3. Download the TeamForge installer from the location provided by your CollabNet representative.
4. Deploy the installation package.

```
rpm -ivh TeamForge-install-5.2.0.0.<build-number>.noarch.rpm
```

 **Note:** To specify the directory where you want the package deployed, add the `--prefix` flag to the `rpm` command.

```
rpm -ivh TeamForge-install-5.2.0.0.<build-number>.noarch.rpm --prefix
/opt/collabnet/teamforge-installer/
```

The installer is now on your server at `/opt/collabnet/teamforge-installer/5.2.0.0`

Configure your CollabNet TeamForge 5.2 site

Identify the host where CollabNet TeamForge 5.2 site will run and the URL where users will find it.


1. With a text editor, open this file:

```
/opt/collabnet/teamforge-installer/5.2.0.0/conf/site-options.conf
```

2. Uncomment and rename the `HOST_localhost` token, replacing `localhost` with the name of the machine on which your TeamForge site will run.
For example, suppose your organization, SuperVillain Inc., has a machine called `appbox.supervillain.org` available for its site to run on.

```
HOST_appbox.supervillain.org=app database subversion cvs
```

The host name is case-sensitive, and must exactly match the hostname that appears when you run the `hostname` command.


 **Tip:** If this is an evaluation site only, you can skip this step. However, some features, such as SSL, will not be available.

3. Uncomment and edit the `DOMAIN` variable to point the site's domain name to the server where the site is running.
Use this format:


```
DOMAIN_<host_name>=<domain_name>
```

For example, suppose SuperVillain Inc. wants the domain name `world domination.supervillain.org` for its development site, which will be hosted on a machine called `appbox.supervillain.org`.

```
DOMAIN_appbox.supervillain.org=world domination.supervillain.org
```

 **Tip:** If this is an evaluation site only, you can skip this step. The site's URL will then be the same as the server's host name. You should configure this variable only if you want the domain name with which users will reach your site to be different from the host name of the server where the application is running.

Your CollabNet TeamForge 5.2 site will be visible at the URL you selected.

 **Tip:** You can customize many other aspects of CollabNet TeamForge 5.2. For more detailed instructions, see [Install an advanced CollabNet TeamForge 5.2 site](#) on page 12.

Install the CollabNet TeamForge 5.2 application

The installer sets up the site according to your configuration settings.

1. Verify that the `sendmail` service is stopped.

If it is running, stop it.

```
/sbin/service sendmail stop
/sbin/chkconfig sendmail off
```

2. Run the installer.

```
cd /opt/collabnet/teamforge-installer/5.2.0.0
./install.sh -a -b -V
```

3. Start the application services.

```
/etc/init.d/collabnet start all
```

Your CollabNet TeamForge 5.2 site is now up and running.


- By default, the URL to log into is the hostname of the server on which your site is installed. If you provided a value for the `DOMAIN` variable in the `site-options.conf` file, then go to that domain instead.
- It's a good idea to stop TeamForge and reboot the machine to make sure all services come up at startup.
- On the site toolbar, click **Admin** > **Users** > **TeamForge Administrator**. If your site's URL has changed, click **Edit** and provide a valid email address. This is the address that appears in the `From` field of messages automatically sent to users from your site.

Supply your TeamForge license key

Your license key enables you to use CollabNet TeamForge for the period of your contract.

Your license key will only work for the IP address of the machine that your CollabNet TeamForge 5.2 is running on, as specified in your order form.

1. Locate the confirmation email you received from your CollabNet representative when you purchased your contract.
2. Log into your site as the site administrator.

 **Note:** The site administrator is different from the root user on the machine where the site is running.

3. Click **Admin** > **License Key**.


If you have entered a license before, the IP address and current licensed number of users on your site are listed on the **License Key** page. Verify that the IP address is the same as the one you entered in your order form.

4. Click **Enter License Key**.

5. Copy your new license key from the confirmation email and paste it into the **Enter License Key** field.

A license key string looks like this:

```
25e9vllnrc14.16.16.5.320257805395867a360c79c830d44d05388b0500a638d48e1325465926766382e3394e3160e332e
```

 **Tip:** save this license key in case you need to reinstall CollabNet TeamForge 5.2.

6. Click **Save**.
7. Verify that the new value for **Licensed Number of Users** matches the total number of licensed users in your contract.

Install an advanced CollabNet TeamForge 5.2 site

Advanced configuration provides the most flexibility and performance for your CollabNet TeamForge site. To set up a site, edit the configuration file and then run the installer.

You can configure each CollabNet TeamForge service to fit your site's particular needs. Each TeamForge service can run on its own server or share a server with one or more other services.

Set up hardware for an advanced CollabNet TeamForge 5.2 installation

Set up and install the hardware that your site will run on.


- For information about the operating systems you can use with CollabNet TeamForge 5.2, see [Supported software for CollabNet TeamForge 5.2](#) on page 66.
- For the hardware requirements, see [Minimum hardware requirements for CollabNet TeamForge 5.2](#) on page 66

You must have root-level access to the machine where you are installing CollabNet TeamForge 5.2

If you anticipate heavy use of your site, you may want to consider installing the database or the source control service on a separate server. See [Set up the database for your CollabNet TeamForge 5.2 site on a separate server](#) on page 38 or [Set up Subversion on its own server](#) on page 38.

1. Install the operating system according to the instructions from the OS provider.
 - On Red Hat 5.3 or CentOS 5.2, do not customize your installation. Take the default settings only.
 - On Red Hat 4.7 or CentOS 4.7, select the Minimal installation option.

2. Make sure port 80 is open.


 **Tip:** For detailed firewall requirements, see [Which ports should I keep open?](#) on page 58

3. Verify that the machine name is resolvable on the network.
 - a) Use the `hostname` command to verify the name of the machine.

```
hostname
bigbox.supervillain.org
```

- b) Use the `nslookup` command to verify that your hostname maps to the right IP address.


```
nslookup bigbox.supervillain.org
Server: 204.16.107.137
Address: 204.16.107.137#53
```

 **Tip:** If there is any doubt about what the system's real IP address is, use the `/sbin/ifconfig` command.

Your server is ready. You can now set up the software environment in which you will install CollabNet TeamForge 5.2.

Get CollabNet TeamForge 5.2


Download the packages you need to install and set up CollabNet TeamForge 5.2.

 **Note:** If you are setting up a CollabNet TeamForge 5.2 site on a network that is not connected to the Internet, see [Install CollabNet TeamForge 5.2 without Internet access](#) on page 19.

1. If you are installing on Red Hat Enterprise Linux 4.7, you must install the yum package manager.
 - a) Download the yum installer from <http://yum.collab.net/yum-install.sh>.
 - b) Run the yum installer.

```
sh yum-install.sh
```


2. Create the directory where you will save the TeamForge installer.

 **Tip:** You can put the installer anywhere, but for simplicity we recommend `/opt/collabnet/teamforge-installer`. All the examples given in these instructions use that path.

```
mkdir -p /opt/collabnet/teamforge-installer
```

3. Download the TeamForge installer from the location provided by your CollabNet representative.
4. Deploy the installation package.

```
rpm -ivh TeamForge-install-5.2.0.0.<build-number>.noarch.rpm
```

 **Note:** To specify the directory where you want the package deployed, add the `--prefix` flag to the `rpm` command.


```
rpm -ivh TeamForge-install-5.2.0.0.<build-number>.noarch.rpm --prefix
/opt/collabnet/teamforge-installer/
```

The installer is now on your server at `/opt/collabnet/teamforge-installer/5.2.0.0` and you are ready to configure your site.

Configure your CollabNet TeamForge 5.2 site

To control the behavior of your CollabNet TeamForge 5.2 installation, edit the master site configuration file.

CollabNet TeamForge 5.2 is controlled by settings in the master configuration file, `site-options.conf`.

 **Note:** This page describes some of the most useful modifications you can make. In pursuit of even greater efficiency and performance under your site's particular circumstances, you may want to set other variables not described here.

1. Create the master configuration file by copying the sample configuration file from the installation package.


```
cd /opt/collabnet/teamforge-installer/5.2.0.0
cp conf/site-options-advanced.conf conf/site-options.conf
```

2. Uncomment and rename the `HOST_localhost` token, replacing `localhost` with the name of the machine on which your TeamForge site will run.

For example, suppose your organization, SuperVillain Inc., has a machine called `appbox.supervillain.org` available for its site to run on.

```
HOST_appbox.supervillain.org=app database subversion cvs
```

The host name is case-sensitive, and must exactly match the hostname that appears when you run the `hostname` command.

 **Tip:** If this is an evaluation site only, you can skip this step. However, some features, such as SSL, will not be available.


3. Uncomment and edit the `DOMAIN` variable to point the site's domain name to the server where the site is running.

Use this format:

```
DOMAIN_<host_name>=<domain_name>
```

For example, suppose SuperVillain Inc. wants the domain name `worlddomination.supervillain.org` for its development site, which will be hosted on a machine called `appbox.supervillain.org`.

```
DOMAIN_appbox.supervillain.org=worlddomination.supervillain.org
```

 **Tip:** If this is an evaluation site only, you can skip this step. The site's URL will then be the same as the server's host name. You should configure this variable only if you want the domain name with which users will reach your site to be different from the host name of the server where the application is running.

4. Under "Database tokens," configure the site to use the database application.
 - If you are using a PostgreSQL database, keep the value of the `DATABASE_TYPE` variable at the default value, `postgresql`.
 - If you are using an Oracle database:
 - Set the value of the `DATABASE_TYPE` variable to `oracle`.
 - Set the value of the `DATABASE_PORT` variable to `1521`.

5. Set values for the other key database variables:

```


DATABASE_NAME=<database_name>
DATABASE_USERNAME=<database_user>
DATABASE_PASSWORD=<database_password>

```


(Replace the names in angle brackets with any name you want.)

6. Specify how you want site-related emails handled by giving values for these variables:

- *SYSTEM_EMAIL*: A valid email address for the system administrator responsible for this site.
- *ADMIN_EMAIL*: A valid email address for the site administrator.
- *JAMES_POSTMASTER_EMAIL*: A valid email address for the person or machine that handles email for the domain, such as `postmaster@supervillain.org`.

 **Tip:** The email accounts specified in the *SYSTEM_EMAIL*, *ADMIN_EMAIL*, and *JAMES_POSTMASTER_EMAIL* variables do not necessarily have to be different from each other.

- *JAMES_GATEWAY_HOST*: A mail server with Internet access. This assures delivery of site email to users if your TeamForge server cannot connect to a DNS server or cannot get outside connections over port 25.


 **Note:** Any email account you specify for the site must be hosted on a separate server from the TeamForge site server.

7. Review the rest of the variables in the `site-options.conf` file to make sure their values are right for your installation, then save the file.

Your site's special requirements are now reflected in its configuration. You can install the software.

Install the CollabNet TeamForge 5.2 application

Run the installer, then update some components and start your site.

 **Tip:** For a direct view of everything the installer is doing, tail the yum log in another terminal before executing these steps.

```
tail -f /var/log/yum.log
```

1. Verify that the `sendmail` service is stopped.

If it is running, stop it.

```

/sbin/service sendmail stop
/sbin/chkconfig sendmail off

```

2. Identify any software that won't work with CollabNet TeamForge 5.2.

```

cd /opt/collabnet/teamforge-installer/5.2.0.0
./environment_check.sh

```

The environment checking utility compares your system environment with the list of packages required by the CollabNet TeamForge 5.2 installer, and reports on which required packages are installed, missing, or out of version tolerance. A package that is out of version tolerance triggers a `WARNING` message. In this case, you have three choices:

1. Remove the package and its dependencies.
2. Have the versions fixed automatically by running the `prepare-environment` script.

```
./prepare-environment.sh
```


The TeamForge installer automatically installs the versions that TeamForge needs.

3. If you can't have packages changed automatically (for example, if some other application depends on the same Subversion or PostgreSQL installation on your application server, or the `prepare-environment.sh` script finds a package conflict it cannot resolve automatically), you can do the upgrade yourself. Use these versions:

- Subversion 1.5.5 (FSFS): To upgrade, see the instructions in [Version Control with Subversion](#).


- PostgreSQL 8.2.12: To upgrade, see the instructions at postgresql.org.

3. Run `environment_check.sh` again until no WARNING messages appear.

 **Important:** Do not move on to the next step until all WARNING messages are resolved.

4. Run the installer.

```
./install.sh -a -V
```

 **Important:** Do not delete the `teamforge-install` directory. You will need it for future maintenance or upgrades.


Set up the database for your CollabNet TeamForge 5.2 site

Install the database and configure TeamForge to work with it.

The site database contains the work products (other than source code) that users will create on your site.


Set up a PostgreSQL database

To use a PostgreSQL database for your CollabNet TeamForge 5.2 data, configure PostgreSQL to work with TeamForge.

 **Note:** These steps are the minimal steps required for a site running on a single server. Settings may vary if your environment features multiple boxes or special networking characteristics.

1. Log in as the PostgreSQL user.

```
su postgres
```

 **Note:** Replace `postgres` with the value you selected for `DATABASE_USERNAME` in the `site-options.conf` file, if you changed it from the default. Your password is the value of the `DATABASE_PASSWORD` in the `site-options.conf` file.

2. In the `/var/lib/pgsql/data/pg_hba.conf` file, make sure the host entry points to your CollabNet TeamForge 5.2 server.

The table should look like this:

```
#TYPE      DATABASE          USER              CIDR-ADDRESS       METHOD
# "local" is for Unix domain socket connections only
local      all                all                trust
# IPv4 local connections:
host       all                all                127.0.0.1/32      trust
# IPv6 local connections:
host       <DATABASE_NAME>  <DATABASE_USERNAME> <app_host_ip>/32  md5
```


where:

- `<DATABASE_NAME>` is the value of the `DATABASE_NAME` variable in the `site-options.conf` file.
- `<DATABASE_USERNAME>` is the value of the `DATABASE_USERNAME` variable in the `site-options.conf` file.
- `<app_host_ip>` is the IP address of the main TeamForge application server (that is, the server to which the `app` parameter is assigned in the `site-options.conf` file).

3. In the `/var/lib/pgsql/data/postgresql.conf` file, under `CONNECTIONS AND AUTHENTICATION`, make sure the `listen_addresses` variable points to your database server.

```
listen_addresses = '127.0.0.1,<database_host_ip>'
```

where `<database_host_ip>` is the IP address of the the server to which the database parameter is assigned in the `site-options.conf` file.

 **Note:** If your database is running on the same machine as the main TeamForge application, point the variable to your application server.

4. Log out of PostgreSQL and restart it.

```
exit

/etc/init.d/postgresql start
```

5. Log back into the PostgreSQL server and create the PostgreSQL user.

```
su postgres
createuser -P -S --createdb --no-createrole <username>
```

6. Create the database.


```
createdb -E UTF8 -O <username> <database name>
```

where <database name> is the value you selected for `DATABASE_NAME` in the `site-options.conf` file.

7. Exit the PostgreSQL shell.

```
exit
```

Your database is now ready to manage the work products your users produce.

 **Tip:** For recommendations on optimizing your PostgreSQL database to fit the particular requirements of your CollabNet TeamForge 5.2 site, see [this wiki page](#).

Set up an Oracle database

To use an Oracle database for your CollabNet TeamForge 5.2 data, set up the Oracle database and tell the installer how to handle it.

1. Make sure your database uses UTF8 or AL32UTF8 encoding.

This is needed to support users in Asian languages.

For information about discovering and changing the database encoding, see [this Oracle knowledge base article](#).

2. Connect to your Oracle database.

```
SQL> connect <adminusername>@<db_name>/<adminpassword> as sysdba
```

3. Create the database user and password you will use to connect from CollabNet TeamForge to Oracle.

```
SQL> create user <sf user> identified by <sf passwd> default tablespace <your
tablespace> temporary tablespace <temporary tablespace>;
```

User created.

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

```
SQL> grant unlimited tablespace to <sf user>;
SQL> grant create snapshot to <sf user>;
SQL> grant create cluster to <sf user>;
SQL> grant create database link to <sf user>;
SQL> grant create procedure to <sf user>;
SQL> grant create sequence to <sf user>;
SQL> grant create synonym to <sf user>;
SQL> grant create trigger to <sf user>;
SQL> grant create type to <sf user>;
SQL> grant create view to <sf user>;
```

```
SQL> grant query rewrite to <sf user>;
SQL> grant alter session to <sf user>;
SQL> grant create table to <sf user>;
SQL> grant create session to <sf user>;
SQL> exit
```

 **Note:** The CollabNet TeamForge installer creates the tables and default values for you.

Start your new TeamForge site

Create the initial data, check your Apache configuration, and start your site.

1. Set up the initial data for the site.

(This is called "bootstrapping" the site.)

```
./bootstrap-data.sh
```

2. Update your Apache configuration.

- a) Back up your existing `/etc/httpd/conf/httpd.conf` file.

```
cd /etc/httpd/conf
```

```
mv httpd.conf httpd.conf.backup
```

- b) Review the new `httpd.conf.cn_new` that was created by the install process.

The `httpd.conf.cn_new` file is an Apache server configuration file that combines your existing Apache configuration with directives specific to CollabNet TeamForge 5.2.

- c) When you are satisfied that `httpd.conf.cn_new` meets all your networking requirements, rename it to `httpd.conf`.

```
mv httpd.conf.cn_new httpd.conf
```

3. Start the application services.

```
/etc/init.d/httpd start
/etc/init.d/postgresql start
/etc/init.d/collabnet start
```

Your CollabNet TeamForge 5.2 site is now up and running.


- By default, the URL to log into is the hostname of the server on which your site is installed. If you provided a value for the `DOMAIN` variable in the `site-options.conf` file, then go to that domain instead.
- It's a good idea to stop TeamForge and reboot the machine to make sure all services come up at startup.
- On the site toolbar, click **Admin** > **Users** > **TeamForge Administrator**. If your site's URL has changed, click **Edit** and provide a valid email address. This is the address that appears in the `From` field of messages automatically sent to users from your site.

Supply your TeamForge license key

Your license key enables you to use CollabNet TeamForge for the period of your contract.

Your license key will only work for the IP address of the machine that your CollabNet TeamForge 5.2 is running on, as specified in your order form.

1. Locate the confirmation email you received from your CollabNet representative when you purchased your contract.
2. Log into your site as the site administrator.

 **Note:** The site administrator is different from the root user on the machine where the site is running.

3. Click **Admin ► License Key**.


If you have entered a license before, the IP address and current licensed number of users on your site are listed on the **License Key** page. Verify that the IP address is the same as the one you entered in your order form.

4. Click **Enter License Key**.

5. Copy your new license key from the confirmation email and paste it into the **Enter License Key** field.

A license key string looks like this:

```
25947111c1416162530025080504567436507436302441045888040026387A0E13576579267663432E349E3160E332E
```

 **Tip:** save this license key in case you need to reinstall CollabNet TeamForge 5.2.


6. Click **Save**.

7. Verify that the new value for **Licensed Number of Users** matches the total number of licensed users in your contract.

Install CollabNet TeamForge 5.2 without Internet access

You can still install CollabNet TeamForge 5.2 even if your local network segment is cut off from the Internet.

The CollabNet TeamForge 5.2 installer automatically connects to the Internet and downloads the software packages needed to run CollabNet TeamForge 5.2. On some highly secure networks, the installer may not be able to reach the Internet. In such cases, you must supply the necessary packages by some other means, such as a CD.

 **Note:** These steps describe only the basic procedure for installing without Internet access. You may need to adapt these steps to fit your organization's security policies.

1. Get the CD with the installation packages from your CollabNet representative.

The CD includes the TeamForge installer and the required auxiliary software.

2. Copy the CD files into `/opt/collabnet/teamforge-installer/5.2.0.0/`

```
mkdir /opt/collabnet/teamforge-installer/5.2.0.0/downloads/
cd /opt/collabnet/teamforge-installer/5.2.0.0/downloads/
cp /root/Disconnected/5.2/CD/<arch>/*.gz
/opt/collabnet/teamforge-installer/5.2.0.0/downloads/
```

3. If you are installing on Red Hat Enterprise Linux 4.7, locate the yum package manager installer on the CD and install it.


```
sh yum-install.sh
```


4. Install the TeamForge installer.

```
rpm -ivh TeamForge-installer-5.2.0.0-<build-number>.noarch.rpm
```

5. Mount the Red Hat installation DVD as `/media/cdrom/`.

```
umount /media/RHEL_5.3\ i386\ DVD/
cd /media/
mkdir cdrom
mount /dev/cdrom ./cdrom/
```

 **Note:** Replace `RHEL_5.3` with the correct release identifier, if necessary.

 **Tip:** If there are any spaces in the automount, unmount it first and mount it as a filepath, with no spaces.

6. Create a yum repository for the CD contents in `/etc/yum.repos.d/`

```
cat /etc/yum.repos.d/cdrom.repo
```

Copy these lines into the file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=1
```

7. Create a `Disconnected` directory and copy the contents of `disconnected.tgz` into it.

```
cp /media/disk/Disconnected.tgz /root/
mkdir /root/Disconnected/
cd /root/Disconnected
tar -xzvf /root/Disconnected.tgz
```

8. Create a yum repository for the CollabNet software.

```
cat /etc/yum.repos.d/collabnet.repo
```

Copy these lines into the file:

```
[CollabNet]
name=collabnet
baseurl=file:///root/Disconnected/5.2/5/$basearch
gpgkey=file:///root/Disconnected/RPM-GPG-KEY-collabnet
enabled=1
gpgcheck=1
```

9. Check to make sure that your configuration files are correct.

Run these commands:

```
yum list httpd
yum list jdk
```

Each command should return one package.

10. Run the installer.

```
./install.sh -a -V
```

11. Under "Location of Help Files" in the `site-options.conf` file, change the value of the `HELP_AVAILABILITY` token to `local`.

This makes TeamForge use the copy of the online help that is installed on the local host. When users click a **Help** link in the application, the resulting help content is served from the application server instead of over the Web.


You can now continue your installation according to the instructions at [Configure your CollabNet TeamForge 5.2 site](#) on page 14.

Install CollabNet TeamForge 5.2 on a virtual machine

To get the functionality of CollabNet TeamForge 5.2 with the ease of installation and maintenance that comes with VMware, run CollabNet TeamForge 5.2 in a VMware player.

Get CollabNet TeamForge 5.2


Download the CollabNet TeamForge 5.2 installer from CollabNet and unzip it on on the machine that will host your TeamForge site.

 **Note:** The machine on which you are running the VMware player must have at least 2 GB RAM and a 2 GHz processor.


1. Download the installer from <http://www.collab.net/products/sfee/tryit.html>.


2. Unzip the `TeamForge_5_2-DL1.zip` file.
3. Install the VMware player.
 - On Windows, double-click the `VMware-player-2.5.1-126130.exe` file.
 - On Linux, run this command:

```
rpm -ivh VMware-Player-2.5.1-126130.i386.rpm
```

 **Tip:** For detailed information about installing VMware on Linux, see the [VMware Player Manual](#).

4. Start the VMware player.
 - On Windows, click **Start > Programs > VMware > VMware Player**.
 - On Linux, run the `vmplayer` executable.

 **Note:** You don't have to update the VMware player, but you can.
5. In the **Commands** section, click **Open**.
6. In the file window, browse to `TeamForge_5_2-DL1\teamforge\image_files\TeamForge.vmx`


 **Note:** Starting the VMware image may take a few minutes, depending on the speed of your system.

Configure CollabNet TeamForge 5.2


After you have installed VMware Player, configure the CollabNet TeamForge 5.2 VMware image.

Only one user needs to configure CollabNet TeamForge 5.2. This instance acts as the application server. To access CollabNet TeamForge 5.2, the CollabNet TeamForge 5.2 application server must be running in VMware Player. Then other users can access it via a Web browser without running VMware Player.


1. Log in with the username `root` and the password `changeme`.
 2. Enter and confirm a new Linux password.

 **Tip:** The system may warn that your password does not meet security standards. For example, it may be too short. This does not mean the password is rejected. If you confirm the same password, it will work.
 3. When prompted to run the configuration tool, type `y`.
 4. Read the product license agreement.


Type `q` to close it.

 **Tip:** You can use the space bar to advance a screen at a time.
 5. If you accept the license terms, type `y`.
 6. In the CollabNet TeamForge 5.2 configuration tool, choose **Dynamic Networking (DHCP)** or **Static Networking (Static IP)**
 - Dynamic networking is useful for a one-person trial installation. It is quick and easy, but email integration with TeamForge will not work correctly.
 - Static networking is best if you are evaluating CollabNet TeamForge 5.2 with a team, or if you already have a license and intend to use TeamForge to support your team.

To configure static networking, you will need to get a static IP address and hostname from your network administrator, and specify your network settings when prompted.

 **Note:** In this case, it's also a good idea to run TeamForge in VMware Player on a dedicated machine.
- The networking for TeamForge is restarted.
7. Specify your outgoing email (SMTP) server.
 - For a one-person evaluation, accept the default value.


- If you have a CollabNet TeamForge 5.2 license and intend to send email outside of your firewall, use the SMTP server settings provided by your network administrator.

 **Note:** Depending on your corporate email configuration, your system administrator may need to permit TeamForge to send mail to the corporate mail server.

8. Choose whether to run CollabNet TeamForge 5.2 at startup.

- Choose “Yes” to start CollabNet TeamForge 5.2 automatically whenever you start the TeamForge VMware image.
- Choose “No” to require a manual CollabNet TeamForge 5.2 startup whenever you start the TeamForge VMware image.

9. At the prompt, click **Enter** to start your CollabNet TeamForge 5.2 site.

 **Note:** Startup can take several minutes, depending on the speed of the host system. On some slower systems, you may get a false failure message from JBoss, like this:

```
jboss (app) (localhost:8080)
.....failed to start in 240 seconds, giving up now.
Please check the
log: /opt/collabnet/teamforge/log/apps/service.log FAILED
```

This can safely be ignored.

10. Log into your new site.

The URL for your site is the IP address or domain name provided in the Linux console at the end of the installation process.


Any user with access to the network where the host system is running can now get to your site via a Web browser.

Supply your TeamForge license key

Your license key enables you to use the VMware edition of CollabNet TeamForge for the period of your contract.

Your license key will only work for the IP address of the machine that your CollabNet TeamForge 5.2 is running on, as specified in your order form.

1. Locate the confirmation email you received from your CollabNet representative when you purchased your contract.
2. Log into your site as the site administrator.

 **Note:** The site administrator is different from the root user on the machine where the site is running.

3. Click **Admin > License Key**.


If you have entered a license before, the IP address and current licensed number of users on your site are listed on the **License Key** page. Verify that the IP address is the same as the one you entered in your order form.

4. Click **Enter License Key**.

5. Copy your new license key from the confirmation email and paste it into the **Enter License Key** field.

A license key string looks like this:

```
25e9vllnrc14.16.16.5.320250805095864360A074KE30244D05B88H0500A6380A8E13254659267663A82E33943160E33E
```

 **Tip:** save this license key in case you need to reinstall CollabNet TeamForge 5.2.

6. Click **Save**.

7. Verify that the new value for **Licensed Number of Users** matches the total number of licensed users in your contract.

Upgrade a CollabNet SourceForge Enterprise 5.1 site to CollabNet TeamForge 5.2

You can upgrade to CollabNet TeamForge 5.2 from CollabNet SourceForge Enterprise 5.1, with any patch installed.

To upgrade from CollabNet SourceForge Enterprise 5.0 or from any version of SourceForge Enterprise Edition, first [upgrade your site to CollabNet SourceForge Enterprise 5.1](#) and start it up. Then follow the steps for upgrading to CollabNet TeamForge 5.2.

Uninstall CollabNet TeamForge 5.1

To upgrade to CollabNet TeamForge 5.2, start by uninstalling CollabNet SourceForge Enterprise 5.1.

The uninstall process removes the SourceForge application, but leaves your site data and configuration in place.

1. Stop your CollabNet SourceForge Enterprise 5.1 site.

```
/etc/init.d/httpd stop
/etc/init.d/postgresql stop
/etc/init.d/collabnet stop
```

2. In the directory where you saved the CollabNet SourceForge Enterprise 5.1 installer, run the uninstall command.

```
cd /usr/local/csfe-install/csfe-install-5.1.0.0.126.i386-redhat-5
./install.sh -n -u -V -d /usr/local/sourceforge
```

The CollabNet SourceForge Enterprise 5.1 installer removes the rpm packages that make up the CollabNet SourceForge Enterprise 5.1 application.

Get CollabNet TeamForge 5.2


Download the packages you need to install and set up CollabNet TeamForge 5.2.

1. If you are installing on Red Hat Enterprise Linux 4.7, you must install the yum package manager.

- a) Download the yum installer from <http://yum.collab.net/yum-install.sh>.
- b) Run the yum installer.

```
sh yum-install.sh
```


2. Create the directory where you will save the TeamForge installer.

 **Tip:** You can put the installer anywhere, but for simplicity we recommend `/opt/collabnet/teamforge-installer`. All the examples given in these instructions use that path.

```
mkdir -p /opt/collabnet/teamforge-installer
```

3. Download the TeamForge installer from the location provided by your CollabNet representative.
4. Deploy the installation package.

```
rpm -ivh TeamForge-install-5.2.0.0.<build-number>.noarch.rpm
```

 **Note:** To specify the directory where you want the package deployed, add the `--prefix` flag to the rpm command.

```
rpm -ivh TeamForge-install-5.2.0.0.<build-number>.noarch.rpm --prefix
/opt/collabnet/teamforge-installer/
```

The installer is now on your server at `/opt/collabnet/teamforge-installer/5.2.0.0`

Get the CollabNet TeamForge 5.2 update package


Download the data migration software to the machine where your upgraded site will run.

1. Download the migration package from the location provided by your CollabNet representative.
2. Unpack the migration packages in the `migration` directory under your installer directory.
For example:

```
cd /opt/collabnet/teamforge-installer/5.2.0.0/migration/
unzip migratorator_5.2.0.0.<build-number>.zip
```

3. Copy the CollabNet SourceForge Enterprise 5.1 configuration file to the new location.

```
cd /opt/collabnet/teamforge-installer/5.2.0.0/conf
cp
/usr/local/csfe-install/csfe-install-5.1.0.0.126.i386-redhat-5/conf/site-options.conf
.
```

 **Note:** If you are moving your site to new hardware, use `scp` instead.

Install the CollabNet TeamForge 5.2 application

Install CollabNet TeamForge 5.2 and bring over the data from your CollabNet SourceForge Enterprise 5.1 site.

1. Verify that the `sendmail` service is stopped.

If it is running, stop it.

```
/sbin/service sendmail stop
/sbin/chkconfig sendmail off
```

2. Identify any software that won't work with CollabNet TeamForge 5.2.

```
cd /opt/collabnet/teamforge-installer/5.2.0.0
./environment_check.sh
```

The environment checking utility compares your system environment with the list of packages required by the CollabNet TeamForge 5.2 installer, and reports on which required packages are installed, missing, or out of version tolerance. A package that is out of version tolerance triggers a `WARNING` message. In this case, you have three choices:

1. Remove the package and its dependencies.
2. Have the versions fixed automatically by running the `prepare-environment` script.

```
./prepare-environment.sh
```

The TeamForge installer automatically installs the versions that TeamForge needs.

3. If you can't have packages changed automatically (for example, if some other application depends on the same Subversion or PostgreSQL installation on your application server, or the `prepare-environment.sh` script finds a package conflict it cannot resolve automatically), you can do the upgrade yourself. Use these versions:

- Subversion 1.5.5 (FSFS): To upgrade, see the instructions in [Version Control with Subversion](#).
- PostgreSQL 8.2.12: To upgrade, see the instructions at postgresql.org.

3. Run the installer.

```
./install.sh -a -V
```

4. Verify that your database is running. If not, start it.

```
/etc/init.d/postgresql start
```

5. Convert your old site's data.

```
cd /usr/local/sourceforge/runtime/scripts
./migrate.py
```

The `migrate.py` script locates the existing site data and modifies it to work with CollabNet TeamForge 5.2.

This includes configuration data for LDAP and the James mail server. Any modifications that you have applied to these components on your CollabNetSourceForge Enterprise 5.1 site are reproduced on your CollabNet TeamForge 5.2 site.

6. Update your Apache configuration.

- a) Back up your existing `/etc/httpd/conf/httpd.conf` file.

```
cd /etc/httpd/conf

mv httpd.conf httpd.conf.backup
```

- b) Review the new `httpd.conf.cn_new` that was created by the install process.

The `httpd.conf.cn_new` file is an Apache server configuration file that combines your existing Apache configuration with directives specific to CollabNet TeamForge 5.2.

- c) When you are satisfied that `httpd.conf.cn_new` meets all your networking requirements, rename it to `httpd.conf`.

```
mv httpd.conf.cn_new httpd.conf
```

7. If you are moving your site to a new machine as part of your upgrade, you must update the file permissions on your site's data.

Use this command to do it automatically:

```
/opt/collabnet/sourceforge/runtime/scripts/fix_data_permissions.sh
```

8. Start the application services.

```
/etc/init.d/httpd start
/etc/init.d/postgresql start
/etc/init.d/collabnet start
```

Your CollabNet TeamForge 5.2 site is now up and running.

Finish your upgrade

Here's how to make sure everything is running smoothly after upgrading to CollabNet TeamForge 5.2.


1. Log into your site as the administrator.

By default, the URL to log into is the hostname of the server on which your site is installed. If you provided a value for the `DOMAIN` variable in the `site-options.conf` file, then go to that domain instead.

2. It's a good idea to stop TeamForge and reboot the machine to make sure all services come up at startup.

3. Synchronize your users' source code access permissions.

- Click **Admin** in the CollabNet TeamForge navigation bar.
- On the site administration navigation bar, click **Integrations**.
- On the **SCM Integrations** tab, select the source control services you want to synchronize.

 **Tip:** It is best to select all of them.

- Click **Synchronize Permissions**.

4. Verify that your site administrator email is still correct.

This is the address that appears in the `From` field of messages automatically sent to users from your site.

- On the site toolbar, click **Admin > Users > TeamForge Administrator**.
- If your site's URL has changed, click **Edit** and provide a valid email address.

5. If your site has custom branding, inspect the `button_bar.vm` and `menu_bar.vm` files in the branding repository and synchronize them with the application defaults.

For more detailed information about branding, see [Change your site's look and feel](#) You can redesign some aspects of your site to suit your organization's needs and preferences. .

6. On the site toolbar, click **Admin > Users > TeamForge Administrator**. If your site's URL has changed, click **Edit** and provide a valid email address.

This is the address that appears in the From field of messages automatically sent to users from your site.

Upgrade a CollabNet SourceForge Enterprise 5.1 site to CollabNet TeamForge 5.2 on a virtual machine

You can upgrade to CollabNet TeamForge 5.2 from a VMware installation of CollabNet SourceForge Enterprise 5.1.

To upgrade from CollabNet SourceForge Enterprise 5.0 or from any version of SourceForge Enterprise Edition, first [upgrade your site to CollabNet SourceForge Enterprise 5.1](#) and start it up. Then follow the steps for upgrading to CollabNet TeamForge 5.2.

Get the CollabNet TeamForge 5.2 upgrade package

Download the software for upgrading your CollabNet TeamForge site from CollabNet.

The machine on which you are running the VMware player must have at least 2 GB RAM and a 2 GHz processor.


 **Important:** If you are already running the free 15-user downloadable version of SourceForge Enterprise Edition 4.4 or earlier, talk to a CollabNet representative before upgrading to CollabNet TeamForge 5.2

1. Download the upgrade packages you need from the location provided by your CollabNet representative.
 - If your current site runs CollabNet TeamForge 5.2 Enterprise Edition 4.4, get the `updater-vmware-4_4-5_1.zip` package.
 - If your current site runs CollabNet TeamForge 5.0, get the `updater-vmware-5_0-5_1.zip` package.

2. Log into your VMware instance as root.

3. In your VMware player, create a directory called `/opt/collabnet/teamforge-install`.

4. Copy the upgrade package into the `/opt/collabnet/teamforge-install` directory in your VMware image.

 **Tip:** A good way to do this is to open a connection to your site's IP address from your local machine with an FTP or SCP client, such as WinSCP. Use the `ifconfig` command from inside your VMware server to find your site's IP address.

5. Unpack the upgrade package.

```
unzip updater-vmware-4_4-5_1.zip
```

Update CollabNet TeamForge 5.2

To upgrade to CollabNet TeamForge 5.2 on VMware, run the automatic updater.

1. Create a backup copy of your existing VMware image.

The easy way to do this is to make a copy of the `image_files` directory from the original install package.

2. If you are upgrading from SourceForge Enterprise Edition 4.4, apply the Service Pack 1 update.

```
cd /opt/collabnet/teamforge-install/updater-vmware-4_4-5_1/update_4.4-sp1
./update-4.4_SP1.sh
```

3. Apply the 5.1 update.

```
cd
/opt/collabnet/teamforge-install/updater-vmware-4_4-5_1/teamforge-install-5.2.0.0.<build-number>.i386-redhat-4
./updater-4_4-5_1.sh --old=/opt/collabnet/teamforge
```

The updater automatically upgrades your existing installation to CollabNet TeamForge 5.2.

Start CollabNet TeamForge 5.2

Launch your VMware site and verify that it is working.

1. Update your Apache configuration, keeping the existing settings around in case you need to roll back any changes.

- a) Back up your existing `/etc/httpd/conf/httpd.conf` file.

```
cp httpd.conf httpd.conf.old
```

- b) Rename `httpd.conf.cn_new` to `httpd.conf`.


```
mv httpd.conf.cn_new httpd.conf
```

2. Create the runtime environment.

```
./install.sh -V -r -d /opt/collabnet/teamforge
```

3. Start the application services.

```
/etc/init.d/httpd start
/etc/init.d/postgresql start
/etc/init.d/collabnet start
```

 **Tip:** If the JBoss server fails to start, kill all Java processes and repeat the startup command.

```
killall java
```

4. Log into the site and verify that things are working.

- By default, the URL to log into is the machine name of the server on which your site is installed. If you provided a value for the `DOMAIN` variable in the `site-options.conf` file, then go to that domain to log in.
- The initial administrator username is `admin` and the password is `admin`. You are required to change the password the first time you log in.
- It's a good idea to stop CollabNet TeamForge 5.2 and reboot the machine to make sure all services come up at startup.

5. Advise your site's users to refresh their browser cache the first time they log into the new site.

Use language like this:

"Welcome to your newly upgraded CollabNet TeamForge 5.2 site. This release comes with significant look and feel improvements designed to make your experience more productive and intuitive. To make sure you are able to view all the new UI elements, please start by pressing Ctrl-F5 to refresh your browser."

 **Tip:** You can use your site's Project News feature to do this. See [Post a news item](#).


6. If you are upgrading to CollabNet TeamForge 5.2 from CollabNet TeamForge 5.0, you must update your site's customizations.

- a) With a source control client such as TortoiseSVN or Subclipse, check out the `branding` repository from the site's `look` project.
- b) Remove the `menu_bar.vm` file from your local working copy.
- c) Commit the changes.


Upgrade to TeamForge 5.2 without Internet access

You can upgrade to CollabNet TeamForge 5.2 from CollabNet SourceForge Enterprise 5.0 or SourceForge Enterprise Edition 4.4 SP1 even if your local network segment is cut off from the Internet.

The CollabNet TeamForge 5.2 installer automatically connects to the Internet and downloads the software packages needed to run CollabNet TeamForge 5.2. However, on some highly secure networks, the install cannot reach the Internet. In such cases, you must supply the necessary packages by some other means, such as a CD.

 **Note:** These steps describe only the basic procedure for upgrading without Internet access. You may need to adapt these steps to fit your organization's security policies.


1. Get on a machine outside your secure network segment.
2. Download the TeamForge installer from the location provided by your CollabNet representative.
3. Download the TeamForge support packages from the location provided by your CollabNet representative.
4. Download the migration package from the location provided by your CollabNet representative.
5. Copy the installer, support packages and migration packages to a CD or other portable medium.
6. Log on as root to the machine where you will install CollabNet TeamForge 5.2.
7. Create the directory where you will save the TeamForge installer.

 **Tip:** You can put the installer anywhere, but for simplicity we recommend `/opt/collabnet/teamforge-installer`. All the examples given in these instructions use that path.

```
mkdir -p /opt/collabnet/teamforge-installer
```

8. Deploy the installation package.

```
rpm -ivh TeamForge-install-5.2.0.0.<build-number>.noarch.rpm
```

 **Note:** To specify the directory where you want the package deployed, add the `--prefix` flag to the `rpm` command.

```
rpm -ivh TeamForge-install-5.2.0.0.<build-number>.noarch.rpm --prefix
/opt/collabnet/teamforge-installer/
```

9. Unpack the migration packages in the migration directory under your installer directory.
For example:

```
cd /opt/collabnet/teamforge-installer/5.2.0.0/migration/
unzip migratorator_5.2.0.0.<build-number>.zip
```

10. Copy the install packages (not the installer) to the `/opt/collabnet/teamforge-installer/5.2.0.0/downloads` directory.

Install a different build of the same release

You can uninstall the current release and install a new build of the same CollabNet TeamForge release without touching your site's data.


Replacing an instance with a new build of the same release on the same hardware is known as "point upgrading."

Point upgrading is a partial application of the process for upgrading to a new release. For comparison, see [Install an advanced CollabNet TeamForge 5.2 site](#) on page 12.

1. Stop CollabNet TeamForge 5.2.

```
/etc/init.d/httpd stop
/etc/init.d/postgresql stop
/etc/init.d/collabnet stop
```

2. Make a copy of the `site-options.conf` file from the old installation.

 **Note:** Don't generate a new one with the `generate-site-options.sh` script.

3. Uninstall CollabNet TeamForge 5.2.

```
cd <installation_source>
./install.sh -u -d <SITE_DIR>
```

4. Remove the original installer package.

```
sudo rpm -e TeamForge-installer-5.2.0.0-180
```


5. Remove the original installer.

```
sudo rm -rf /var/ops/teamforge-installer
```

6. Download the TeamForge installer from the location provided by your CollabNet representative.

7. Deploy the installation package.

```
rpm -ivh TeamForge-install-5.2.0.0.<build-number>.noarch.rpm
```

 **Note:** To specify the directory where you want the package deployed, add the `--prefix` flag to the `rpm` command.

```
rpm -ivh TeamForge-install-5.2.0.0.<build-number>.noarch.rpm --prefix
/opt/collabnet/teamforge-installer/
```

8. Deploy the new installer.

```
sudo rpm -ivh --prefix=/var/ops
/home/svd/TeamForge-installer-5.2.0.0-181.noarch.rpm
```

9. Copy the `site-options.conf` file into the new `/conf` directory.

10. Start the application services.

```
/etc/init.d/collabnet start all
```

Chapter

2

Support CollabNet TeamForge users

Topics:

- *Authenticate users with LDAP*
- *Let users see what's in a project template*

Authenticate users with LDAP

Use LDAP to facilitate managing users and groups in CollabNet TeamForge 5.2.

Set up LDAP integration for the CollabNet TeamForge 5.2 server

Follow these steps to convert your CollabNet TeamForge 5.2 installation to authenticate against your corporate LDAP server.

1. Shut down CollabNet TeamForge 5.2.

```
/etc/init.d/httpd stop
/etc/init.d/postgresql stop
/etc/init.d/collabnet stop
```

2. Copy the LDAP configuration file to the data directory.

```
cd <SITE_DIR>
cp dist/jboss/jboss-3.2.6/server/default/conf/login-config.xml
var/etc/login-config.xml
```

3. Edit the `<installation_source>/conf/site-options.conf` file.

- a) Tell CollabNet TeamForge 5.2 to use LDAP authentication.

Under "External User Authentication," uncomment this line:


```
USE_EXTERNAL_USER_AUTHENTICATION=false
```

and change its value to true.

- b) Tell CollabNet TeamForge 5.2 where to look for your LDAP configuration settings.


Uncomment this line:

```
LOGIN_CONFIG_XML_FILE={__DATA_DIR__}/etc/login-config.xml
```

 **Note:** `DATA_DIR` is usually mapped to the `<SITE_DIR>/var` directory. You may want to check the `SITE_DIR` and `DATA_DIR` variables.

- c) Check that the `MINIMUM_PASSWORD_LENGTH` variable matches the limit used on the LDAP server.

If your LDAP server does not enforce a minimum password length, set `MINIMUM_PASSWORD_LENGTH` to 0 (zero).

 **Note:** If a password is used in LDAP that is shorter than the minimum allowable password length in CollabNet TeamForge 5.2, you will not be able to create the user in CollabNet TeamForge 5.2.


4. In the `<SITE_DIR>/var/etc/login-config.xml` file, modify the SourceForge `application-policy` block to enable CollabNet TeamForge 5.2 to authenticate against your LDAP server.

 **Tip:** The `application-policy` block begins on line 113 of the `login-config.xml` file.


- a) Replace the SourceForge `application-policy` block with the code listed in the sample `application-policy` block in [login-config.xml](#) on page 80.

- b) Replace `principalDNPrefix` with your LDAP username parameter.

In the example `application-policy` block, the username is stored in LDAP as the `uid` parameter.


 **Note:** Make sure to include the trailing `=` in the prefix.

- c) Replace `principalDNSuffix` with the LDAP domain in which usernames are stored.

 **Note:** Make sure to include the leading comma in the suffix if one is needed.

- d) 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 `ldap://util.dev.sf.net:389/`.

 **Note:** Make sure to include `ldap://` at the beginning of the URL.

5. Save all the files you have edited and change their ownership back to `sf-admin`.

```
chown sf-admin:sf-admin login-config.xml
```


6. Recreate the runtime environment.

```
./install.sh -V -r -d <SITE_DIR>
```

Set up LDAP for a source control integration server

Using LDAP on an integration server can speed up performance for users.

By default, a source code integration server authenticates users via UNIX users and groups. However, managing a large number of users and repositories this way can be slow. To reduce the time required to create, manipulate, and synchronize users and groups, configure the integration server to use LDAP.


 **Note:** Do not use this technique to connect an integration to a pre-existing corporate LDAP system. This approach uses a local, private LDAP server to replace `/etc/passwd` and `/etc/group` for user and group management.

Take the following steps on each integration server you want to convert to LDAP.

1. Ensure that the following RPMs are installed:


- `openldap`
- `openldap-clients`
- `openldap-servers`

2. Verify that `/etc/openldap/cacerts` exists.

 **Note:** In RHEL 4, installation of `openldap` may not create this directory, which is required by `openldap` to start.

3. In the `/etc/openldap/slapd.conf` file, change the following values:

Option	Value
<code>suffix</code>	<code>"dc=CollabNet TeamForge 5.2,dc=com"</code>
<code>rootdn</code>	<code>"cn=Admin,dc=CollabNet TeamForge 5.2,dc=com"</code>
<code>rootpw</code>	<code>sfee</code>
<code>sizelimit</code>	<code>30000</code>

 **Tip:** To supply an encrypted password instead of clear text for `rootpw`, run the command

```
slappasswd
```

You are prompted for a password, and a string that looks like the following is displayed:

```
{SSHA} 7hC2H50oEZ0aT6rL3hAvyxy11jrZYB2
```

Use that string instead of the `sfee` used in the `rootpw` example.

4. Configure the LDAP service to start on boot.

```
chkconfig --level 345 ldap on
```


5. Make sure the ldap database directory is clean and has the correct permissions.

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

6. Start the LDAP server.

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

7. Create an initial LDIF (LDAP Interchange Format) file for your groups.

 **Important:** Make sure no lines in the ldif content start with white space.

Refer to the following for ldif content:

```
dn: dc=CollabNet TeamForge 5.2,dc=com dc: CollabNet TeamForge 5.2
objectClass: top objectClass: domain dn: ou=Users,dc=CollabNet TeamForge
5.2,dc=com ou: Users objectClass: organizationalUnit dn:
ou=Groups,dc=CollabNet TeamForge 5.2,dc=com ou: Groups objectClass:
organizationalUnit dn: cn=sfee,ou=Groups,dc=CollabNet TeamForge 5.2,dc=com
cn: sfee objectClass posixGroup gidNumber: 30000
```

8. Add the ldif information.

```
ldapadd -x -DÂ cn=Admin,dc=CollabNet TeamForge 5.2,dc=comÂ -W < initial.ldif
```

When you are prompted for a password, type the password you used in `slapd.conf`.


9. Manually remove groups and users created by CollabNet TeamForge from `/etc/group` and `/etc/passwd`.

These are usually grouped at the end of the respective files. The groups include `sfall`, `sfunrest`, and all groups named `reps` with a numeric suffix (for example: `reps1001`).


 **Note:**

- If users are left in the `/etc/passwd` or `/etc/shadow` files, those users may get permission errors when committing code to repositories on that integration server.
- If groups created by CollabNet TeamForge 5.2 are left in the `/etc/group` and `/etc/gshadow` files, users of those groups may get permission errors when checking in. If groups are removed from these files, a synchronize external system call will be required to restore correct permissions.

10. Configure the server to authenticate from LDAP.

 **Tip:** On RHES, you can use the `authconfig` command. If you have a valid display defined, this command will pop up an X window. Otherwise you can use the command line interface.

- a) Select **Use LDAP**.
- b) Specify the base DN and server.

 **Note:** Do not select **Use TLS**.

- c) Click **Next**.
- d) Select **Use LDAP Authentication**. Fill in the LDAP basedn and host information.
- e) Click **OK**.

This program modifies some `/etc/pam.d` entries and writes out a valid `/etc/ldap.conf` file.

11. Restart sshd.

```
/etc/init.d/sshd restart
```

12. In `/etc/nsswitch.conf`, verify that the `passwd`, `shadow`, and `group` entries look like this:

```
passwd: files ldap shadow: files ldap group: files ldap
```

13. Verify that `/etc/ldap.conf` contains these values:

```
nss_base_passwd ou=Users,dc=CollabNet TeamForge 5.2,dc=com?one nss_base_shadow
ou=Users,dc=CollabNet TeamForge 5.2,dc=com?one nss_base_group
ou=Groups,dc=CollabNet TeamForge 5.2,dc=com?one pam_filter
objectClass=posixAccount pam_login_attribute uid
```


14. Log onto the integration server.
15. To configure the CollabNet TeamForge integration server to update LDAP, add these entries to `/conf/site-options.conf`:

Entry	Value	Description
CVS_LDAP_HOST	localhost	The host of the LDAP server.
CVS_LDAP_PORT	389	The port of the LDAP server.
CVS_LDAP_USERS_DN	Example: ou=Users,dc=CollabNet TeamForge 5.2,dc=com)	Schema address that users are added into.
CVS_LDAP_GROUPS_DN	Example: ou=Groups,dc=CollabNet TeamForge 5.2,dc=com	Schema address that groups are added into.
CVS_LDAP_USERS_STARTID	30000	The numeric userid to start counting from when creating new users.
CVS_LDAP_GROUPS_STARTID	30005	The numeric groupid to start counting from when creating new groups.
CVS_LDAP_BIND_DN	Example: cn=Admin,dc=CollabNet TeamForge 5.2,dc=com	The "root dn" for the LDAP server. This must be the value specified in <code>slapd.conf</code> .
CVS_LDAP_BIND_PASSWORD		The password for the root dn.
INTEGRATION_OS	linux_ldap	
CVS_USER_DEFAULT_GROUP	sfee	

16. Restart the integration server.


```
<SITE_DIR>/runtime/scripts/CollabNet TeamForge 5.2-integration-init.sh
restart
```

17. Log into the CollabNet TeamForge site's web interface as site administrator.
18. On the **Integration Systems** page, synchronize permissions for all managed source code integration servers.

 **Note:** 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 SourceForge password). The password cannot be set automatically during migration because only the encrypted version is available.

19. When synchronize permissions has completed, correct the permissions on the home directories of your users.
On each source code server that you have converted to LDAP, run these commands:

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

 **Note:** You may see some errors. This is normal and indicates disabled/deleted users.


Modify the application policy


To enable CollabNet TeamForge 5.2 to authenticate against your LDAP server, modify the application-policy block of the `login-config.xml` file.

When the username is passed to the login module from CollabNet TeamForge 5.2, it is translated into a DN for lookup on the LDAP server.

1. The DN that is sent to the LDAP server is:

```
<principalDNPrefix><username><principalDNSuffix>
principalDNPrefix - Replace principalDNPrefix with your LDAP
username parameter.
```


 **Note:** In the example application-policy block, the username is stored in LDAP as the uid parameter.

 **Important:** Be sure to include the trailing = in the prefix.

2. `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:


```
,ou=People,dc=dev,dc=sf,dc=net
```


 **Important:** Be sure to include the leading comma in the suffix if one is needed.

3. 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:

```
ldap://util.dev.sf.net:389/
```


 **Note:** Be sure to include `ldap://` at the beginning of the URL.

 **Important:** To complete your CollabNet TeamForge 5.2 configuration and enable your CollabNet TeamForge 5.2 JBoss installation to authenticate against your corporate LDAP server, you must restart CollabNet TeamForge 5.2.

Let users see what's in a project template

Help your site's project administrators choose a project template by enabling them to see the contents of the templates that are available.

Project administrators normally create a project from a project template. To choose the right template, it may help if they can find out if tasks, documents, wiki pages or other kinds of content are included in a given template. As a site administrator, you can make this possible.

 **Note:** By default, site administrators can see project template details, but other users cannot.

1. Open the `conf/site-options.conf` file in a text editor.
2. Change the value of the `sf.showProjectTemplateDetailToNonSiteAdmins` variable to `true` and save `conf/site-options.conf`.
3. Recreate the runtime environment.

```
./install.sh -V -r -d <SITE_DIR>
```

Chapter

3

Grow your CollabNet TeamForge installation

Topics:

- *Set up Subversion on its own server*
- *Set up the database for your CollabNet TeamForge 5.2 site on a separate server*

Set up Subversion on its own server


To make Subversion code repositories available for your site, install Subversion on a server and set up CollabNet TeamForge 5.2 to use it.

1. On the host that is to be your Subversion server, download the TeamForge 5.2 installer package.

See [Get CollabNet TeamForge 5.2](#) on page 13 for instructions.

2. Deploy the installation package.

```
rpm -ivh TeamForge-install-5.2.0.0.<build-number>.noarch.rpm
```

 **Note:** To specify the directory where you want the package deployed, add the `--prefix` flag to the `rpm` command.

```
rpm -ivh TeamForge-install-5.2.0.0.<build-number>.noarch.rpm --prefix
/opt/collabnet/teamforge-installer/
```

3. On the host where your main TeamForge application is running, open the `site-options.conf` file.
4. Under "Application settings" in the `site-options.conf` file, create (or reuse) a `HOST_<hostname>` token to specify the machine where the source code management service will run.

For example:

```
HOST_codebox.supervillain.org=subversion
```

 **Note:** You can specify `subversion`, `cvs`, or `perforce`.

5. Edit the existing `HOST` variable to reflect the fact that your source control service is no longer running on that machine. For example, suppose your organization, SuperVillain Inc., has a machine called `appbox.supervillain.org` available for its site to run on and a machine called `codebox.supervillain.org` for its Subversion services. Your `site-options.conf` file will now look like this:

```
HOST_appbox.supervillain.org=app database
HOST_codebox.supervillain.org=subversion
```

6. Copy the `site-options.conf` file to the `teamforge-installer/5.2.0.0/conf` directory on the source code machine.

```
scp conf/site-options.conf
<subversion_server>://opt/collabnet/teamforge-installer/5.2.0.0/conf/
scp conf/site-options.conf
<cvs_server>://opt/collabnet/teamforge-installer/5.2.0.0/conf/
```

7. Install TeamForge on your source control server as instructed in [Install the CollabNet TeamForge 5.2 application](#) on page 15 (skipping the instructions for setting up a database).

Your users can now create and manage Subversion repositories from inside their TeamForge site.


Set up the database for your CollabNet TeamForge 5.2 site on a separate server

If you expect your site to have a high rate of user traffic, you may want to run the site's database on its own server.

The database is where users' project pages, documents, tracker artifacts, tasks, discussions and other work products are stored and accessed.


To run your TeamForge site's database on its own server, you must install the database and configure TeamForge to work with it.

The advantage of hosting a service on a separate server is that it does not share CPU, RAM or I/O bandwidth with the server that is hosting the main TeamForge application.

 **Note:** Each TeamForge site can have only one database server.

Set up a PostgreSQL database on its own server

To use a PostgreSQL database for your CollabNet TeamForge 5.2 data, install and configure the database server and set up TeamForge to use it.

 **Tip:** Gather the IP addresses of the machines where you are installing the TeamForge application and database. You will need them as you follow the steps on this page.

1. Download and install PostgreSQL on the host that will be your standalone database server.
(May we suggest these [PostgreSQL install instructions](#).)
2. Configure PostgreSQL to work with your site.
See [Set up a PostgreSQL database](#) on page 16 for instructions.
3. Go to the host where your TeamForge application is running.
4. Under "Application settings" in the `site-options.conf` file, create (or reuse) a `HOST_<hostname>` token to specify the machine where the database service will run.
For example:


```
HOST_databox.supervillain.org=database
```

5. Under "Database tokens," configure the site to use the database application.
 - If you are using a PostgreSQL database, keep the value of the `DATABASE_TYPE` variable at the default value, `postgresql`.
 - If you are using an Oracle database:
 - Set the value of the `DATABASE_TYPE` variable to `oracle`.
 - Set the value of the `DATABASE_PORT` variable to `1521`.
6. Set values for the other key database variables:

```
DATABASE_NAME=<database_name>
DATABASE_USERNAME=<database_user>
DATABASE_PASSWORD=<database_password>
```

(Replace the names in angle brackets with any name you want.)

Your database is now ready to manage the work products your users produce.

 **Tip:** For recommendations on optimizing your PostgreSQL database to fit the particular requirements of your CollabNet TeamForge 5.2 site, see [this wiki page](#).

Set up an Oracle database on its own server

To use an Oracle database for your CollabNet TeamForge 5.2 data, set up the Oracle database and tell the installer how to handle it.

1. Install Oracle according to the instructions provided.
2. Under "Application settings" in the `site-options.conf` file, create (or reuse) a `HOST_<hostname>` token to specify the machine where the database service will run.
For example:

```
HOST_databox.supervillain.org=database
```

3. Under "Database tokens," configure the site to use the database application.

- If you are using a PostgreSQL database, keep the value of the `DATABASE_TYPE` variable at the default value, postgresql.
- If you are using an Oracle database:
 - Set the value of the `DATABASE_TYPE` variable to oracle.
 - Set the value of the `DATABASE_PORT` variable to 1521.

4. Set values for the other key database variables:

```
DATABASE_NAME=<database_name>
DATABASE_USERNAME=<database_user>
DATABASE_PASSWORD=<database_password>
```

(Replace the names in angle brackets with any name you want.)

5. Make sure your database uses UTF8 or AL32UTF8 encoding.

This is needed to support users in Asian languages.

For information about discovering and changing the database encoding, see [this Oracle knowledge base article](#).

6. Connect to your Oracle database.

```
SQL> connect <adminusername>@<db_name>/<adminpassword> as sysdba
```


7. Create the database user and password you will use to connect from CollabNet TeamForge to Oracle.

```
SQL> create user <sf user> identified by <sf passwd> default tablespace <your
tablespace> temporary tablespace <temporary tablespace>;
```

User created.

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

```
SQL> grant unlimited tablespace to <sf user>;
SQL> grant create snapshot to <sf user>;
SQL> grant create cluster to <sf user>;
SQL> grant create database link to <sf user>;
SQL> grant create procedure to <sf user>;
SQL> grant create sequence to <sf user>;
SQL> grant create synonym to <sf user>;
SQL> grant create trigger to <sf user>;
SQL> grant create type to <sf user>;
SQL> grant create view to <sf user>;
SQL> grant query rewrite to <sf user>;
SQL> grant alter session to <sf user>;
SQL> grant create table to <sf user>;
SQL> grant create session to <sf user>;
SQL> exit
```

 **Note:** The CollabNet TeamForge installer creates the tables and default values for you.

Chapter

4

Protect your CollabNet TeamForge site

Topics:


- [Set up SELinux](#)
 - [Protect Apache with SSL](#)
 - [Protect integrations with SSL](#)
 - [Set up SSH tunneling](#)
-

Set up SELinux

If SELinux is running, modify it to allow the services that CollabNet TeamForge 5.2 requires.

1. Enable Apache (running on port 80) to proxy traffic to JBoss (running on port 8080).

```
setsebool -P httpd_can_network_connect 1
```

 **Note:** If you are installing on CentOS 4.6, skip this step.

2. Change the context for your Subversion source code service.

```
chcon -R -h -t httpd_sys_content_t /svnroot
```

3. Change the context of the Subversion repository that handles the branding (look and feel) of your site.

```
chcon -R -h -t httpd_sys_content_t /sf-svnroot
chcon -R -h -t httpd_sys_content_t <SITE_DIR>/var/overrides
```

4. Change the context for your local CVS repository, if you have one.

```
chcon -R -h -t httpd_sys_content_t /cvsroot
```

Protect Apache with SSL

Use Secure Socket Layer (SSL) to run your Web server securely.

Set up Apache for SSL encryption


To force all CollabNet TeamForge 5.2 traffic to use SSL encryption (HTTPS), state that preference in your configuration file.

1. Back up your existing `/etc/httpd/conf/httpd.conf` file.
2. Update the `/opt/collabnet/teamforge-installer/5.2.0.0/conf/site-options.conf` file to support SSL.
 - a) Set the value of the `SSL` variable to `on`.
 - b) Set the value of the `SSL_CERT_FILE` variable to the location of the file that contains your site's SSL certificates.

```
SSL_CERT_FILE=www.example.com.crt
```

- c) Set the value of the `SSL_KEY_FILE` variable to the location of the file that contains your site's RSA private keys.

```
SSL_KEY_FILE=www.example.com.key
```

 **Important:** Select a location for your cert file and your key file that is permanent across restarts. Don't use a temp directory that can be wiped out.

3. Recreate the runtime environment.

```
./install.sh -V -r -d <SITE_DIR>
```

4. Rename the `httpd.conf.cn_new` file to `httpd.conf` and restart the Apache service.

When you point your browser at CollabNet TeamForge 5.2, it should now automatically redirect to HTTPS traffic.

Generate Apache SSL certificates

To use https for web traffic, you will need to obtain a valid Apache SSL certificate.


When generating an Apache (mod_ssl) SSL certificate, you have two options:

- Purchase a SSL certificate from a certificate authority (CA). Searching the Web for "certificate authority" will present several choices.
- Generate 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, this option can be a mild annoyance to some users, because Internet Explorer (IE) issues a harmless warning each time a user visits a site that uses a self-signed certificate.

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

1. Know the fully qualified domain name (FQDN) of the website for which you want to request a certificate.


If you want to access your site through `https://www.example.com`, then the FQDN of your website is `www.example.com`.

 **Note:** This is also known as your common name.

2. Generate the key with the SSL `genrsa` command.

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


This command generates a 1024 bit RSA private key and stores it in the file `www.example.com.key`.

 **Tip:** Back up your `www.example.com.key` file, because without this file your SSL certificate will not be valid.

3. Generate the CSR with SSL `req` command.

```
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. Give the fully qualified domain name, such as `www.example.com`, when prompted for Common Name.

 **Note:** 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.

4. Generate a self-signed certificate.

```
openssl x509 -req -days 370 -in www.example.com.csr -signkey
www.example.com.key -out www.example.com.crt
```

This command will generate a self-signed certificate in `www.example.com.crt`.


You will now have an 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.

Prevent HTTPS cracking

To reduce the risk of HTTPS ciphers being cracked, allow only the strongest ciphers available.

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.

1. In the `/etc/httpd/conf.d/ssl.conf` file, find the headings `SSLProtocol` and `SSLCipherSuite`.

 **Note:** If they do not exist, add them below the `SSLEngine` line.

- In each section, add the following two lines:

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

- Save the file and restart Apache:

```
apachectl restart
```

Protect integrations with SSL


If you have registered Secure Socket Layer (SSL) certificates, your site's users can use SSL when they set up an SCM integration server.

If you use certificates that are generated in-house, self-signed, or signed by a non-established Certificate Authority, they must be registered with each client system that will connect to the CollabNet TeamForge 5.2 server. Registration consists of importing custom certificates into the Java runtime's global keystore on each server.

 **Important:** This will affect any other Java applications on the server that use the same Java runtime.

- Collect server certificates from all servers.

On RHEL, CentOS and other RedHat-based distributions, these are contained in
`/etc/httpd/conf/ssl.crt/server.crt`.

 **Tip:** Be sure to use exactly this path, as there are other files with similar names, plus server certificates are not really secret, but some other files are. So, files must be copied (e.g., via scp) to the same directory, and renamed if necessary to avoid clashes. We recommend that you use the short server name of the corresponding server for this.

- Locate the Java keystore.

This is `PATH_TO_JAVA/jre/lib/security/cacerts`.

For example, this may be `/usr/local/j2sdk1.4.2_10/jre/lib/security/cacerts`.

- Locate the Java keytool utility.

This is `PATH_TO_JAVA/bin/keytool`

For example, `/usr/local/j2sdk1.4.2_10/bin/keytool`.

- Import each server certificate into the keystore.


```
PATH_TO_JAVA/bin/keytool -import -keystore
PATH_TO_JAVA/jre/lib/security/cacerts -file <server>.crt -alias <server>
```

- At the password prompt, use `changeit`.

Confirm that you trust the certificate by typing `yes`.

- Verify that all your certificates are added.

```
PATH_TO_JAVA/bin/keytool -list -keystore
PATH_TO_JAVA/jre/lib/security/cacerts |less
```

 **Note:** The list will contain many more certificates. These are top-level CA certificates, provided with Java.

- Update `/etc/sourceforge.properties` to enable secure communication.

- Set `sfmain.integration.listener_ssl` to true.
- Set `sfmain.integration.listener_port` to 443.

- If you are running more than one separate server, repeat these steps for each server.

- Restart TeamForge

Now you can check the **Use SSL** checkbox when creating an SCM integration.

Set up SSH tunneling

To enable users to access your site through an SSH tunnel, configure the site to allow SSH and then create an SSH key.

1. In the `/opt/collabnet/teamforge-installer/5.2.0.0/conf/site-options.conf` file, set the value of the `SSH_TUNNEL_ENABLED` variable to true.

2. Recreate the runtime environment.

```
./install.sh -V -r -d <SITE_DIR>
```

3. Set the tunnel password.

```
passwd tunnel
```

4. Generate the SSH key.

```
ssh-keygen -d -C {your email}
```

5. Install the `id_dsa.pub` file on the box you want to ssh to.

```
cd ~tunnel (/opt/sourcecast/data/home/tunnel)
-s
cd .ssh
cat > authorized_keys2
```

Chapter

5

Maintain your CollabNet TeamForge installation

Topics:

- *Monitor services on your site*
- *Get information about a CollabNet TeamForge 5.2 site*
- *Patch CollabNet TeamForge 5.2*
- *Specify DNS servers*
- *Optimize PostgreSQL with vacuum*
- *Change the location of a log file*
- *Back up and restore CollabNet TeamForge 5.2 data*

Monitor services on your site

To monitor a CollabNet TeamForge 5.2 service, activate the monitoring script for that service.

Services you can monitor include Apache, Tomcat, JBoss and PostgreSQL.

1. Set up the Big Brother client on your CollabNet TeamForge 5.2 server.

 **Note:** Do this only if the Big Brother client is not already installed.

- a) Set the *BBHOME* environment variable to `/opt/bb/bb`.


```
export BBHOME=/opt/bb/bb
```

- b) Install the Big Brother client at that location.

2. Copy the script for the service you want to monitor into the `BBHOME/ext` directory.
3. Add an entry about this script in the `BBHOME/etc/bb-bbexttab` file.

Such an entry will look like this:

```
localhost : : raid.sh bb-memory.sh sys.sh bb-check-jboss.sh bb-check-tomcat.sh
```

 **Note:** Entries related to the Big Brother server itself should be added in the `BBHOME/etc/bb-hosts` file.

Get information about a CollabNet TeamForge 5.2 site

Use the `snapshot.py` utility to determine what processes are running on your CollabNet TeamForge 5.2 site, how much free memory is available, and other information.

1. Log into the server.
2. Find the application in distress.
3. Run the `snapshot.py` script.

```
<SITE_DIR>/runtime/scripts/snapshot.py
```

Snapshot gathers data from several processes running on the system, including:

- JBoss
- Tomcat
- James
- PostgreSQL
- Apache

The information is written to `LOG_DIR/runtime/snapshot.log` and `LOG_DIR/apps/server.log`.

 **Note:** `LOG_DIR` is the directory you defined as the logging directory in the `site-options.conf` file.

Patch CollabNet TeamForge 5.2

When new functionality or bug fixes must be applied to a running CollabNet TeamForge 5.2 site, you may need to install or remove a patch.


Before doing anything with patches, locate the directory where you saved the CollabNet TeamForge 5.2 installer.

 **Note:** For convenience, this is referred to as the installation source directory.

By default, the installation source directory is `/opt/collabnet/teamforge-installer/5.2.0.0`. If necessary, download the installer from the CollabNet web site and unpack it in this directory.

Go to an arbitrary patch level


Use the `upgrade-site.sh` script to upgrade or downgrade to a patch level of your choosing.

 **Note:** If your CollabNet TeamForge 5.2 site is running on multiple servers, follow this procedure on all of them.

1. Download the patch bundle into the installation directory.
2. Unpack the patch bundle.

```
tar xzvf <name of tar file>
```

3. Stop CollabNet TeamForge 5.2.

 **Note:** If your site is running on multiple machines, stop all the machines.

```
/etc/init.d/collabnet stop all
```

4. Specify the patch level you want to get to in one of these ways:

- a) Use the level option.

```
./upgrade-site.sh -d <SITE_DIR> -l <level>
```

where `<level>` is the patch level you are upgrading or downgrading to.

For example, if your site has patch level 3 and you want to get it to patch level 1, use this command:

```
./upgrade-site.sh -d /opt/collabnet/teamforge -l 1
```

- b) Use the manifest option to point to a manifest file that specifies the patch level you want.

```
./upgrade-site.sh -d <SITE_DIR> -f <manifest>
```

where `<manifest>` is the appropriate manifest file for the patch level you are installing.

For example, if your site has patch level 3 and you want to get it to patch level 1, use this command:

```
./upgrade-site.sh -d /opt/collabnet/teamforge -f manifest-1
```

5. Start CollabNet TeamForge 5.2 on all the machines in the group.

```
/etc/init.d/httpd start
/etc/init.d/postgresql start
/etc/init.d/collabnet start
```

6. Verify that the patch was removed.

- a) Log onto the site as a site administrator.
- b) On the **Admin** tab, click **System Tools**.
- c) Click **Build Information** and observe which patches are present.

Revert a patch upgrade or downgrade

Use the rollback option of the `upgrade-site.sh` script to revert the system's patch level to whatever it was before your last action.


For example, if you just downgraded from patch level 4 to patch level 3, you can use the rollback to bring the site back to patch level 4.

Contrariwise, if you just upgraded from patch level 1 to patch level 3, you can use the rollback option to revert to patch level 1.

1. Change to the `/opt/collabnet/teamforge-installer/5.2.0.0` directory.

```
cd /opt/collabnet/teamforge-installer/5.2.0.0
```

2. Stop CollabNet TeamForge 5.2.

 **Note:** If your site is running on multiple machines, stop all of them.

```
/etc/init.d/collabnet stop all
```

3. On each machine in the group methods, run the `rollback` command.

```
./upgrade-site.sh -d <SITE_DIR> -r
```

where `<SITE_DIR>` is the installation directory where the application is already installed.

4. Start CollabNet TeamForge 5.2 on all the machines in the group.

```
/etc/init.d/httpd start
/etc/init.d/postgresql start
/etc/init.d/collabnet start
```

5. Verify that the patch change was reverted.


- a) Log into the site as a site administrator.
- b) On the **Admin** tab, click **System Tools**.
- c) Click **Build Information** and observe which patches are present.

Remove a patch

Use the `uninstall` option of the `upgrade-site.sh` script to decrement the patch level to the next lowest version.

For example, use this procedure if your site has patch level 3 and you want to bring it back to patch level 2.

1. Stop CollabNet TeamForge 5.2.

 **Note:** If CollabNet TeamForge 5.2 is running on multiple machines, stop all of them.

```
/etc/init.d/collabnet stop all
```

2. Use the `uninstall` option to get to a level that is one less than the current patch level.

```
./upgrade-site.sh -d <SITE_DIR> -u
```

3. Start CollabNet TeamForge 5.2 on all machines in the group.

```
/etc/init.d/httpd start
/etc/init.d/postgresql start
/etc/init.d/collabnet start
```

4. Verify that the patch was removed.

- a) Log onto the site as a site administrator.
- b) On the **Admin** tab, click **System Tools**.
- c) Click **Build Information** and observe which patches are present.

Troubleshoot patches

You may encounter problems like these when applying or removing a patch.

Can't install patch

When you try to install a patch, the component upgrade process may error out if you do not have a full component upgrade bundle.

You may see the following error:

```
Error message: "The patch file <package name> could not be found in the patch
directory. Please verify the patch and try again."
```

1. Check that the RPMs in the install list are present in the upgrade component directory.
2. If you receive this message you should verify that you have received and unpacked the whole Component Upgrade bundle.

Patch installation fails


If the environment is corrupted, the patch installation process may fail.

Suppose you are upgrading an installation from patch level 1 to level 4 and the system has finished uninstalling the packages. While trying to install the first package, the system encounters a problem and the installation fails.

Run the `upgrade-site.sh` script with the `-F` option to get past a failed installation.

In this scenario, you can get to patch level 1 with one of these commands:

- `./upgrade-site.sh -d /opt/collabnet/teamforge -r -F`
- `./upgrade-site.sh -d /opt/collabnet/teamforge -l 1 -F`
- `./upgrade-site.sh -d /opt/collabnet/teamforge -f manifest-1 -F`

 **Note:** The `-F` option forces the specified upgrade or downgrade.

Specify DNS servers

Define the DNS servers with which you want CollabNet TeamForge 5.2 to resolve URLs by listing them in the `resolv.conf` file.

1. In the `/etc/resolv.conf` file, list the servers you want to use for resolving Internet addresses.
2. Rebuild the runtime environment.

```
./install.sh -V -r -d <SITE_DIR>
```

3. Restart CollabNet TeamForge 5.2.

```
/etc/init.d/httpd start
/etc/init.d/postgresql start
/etc/init.d/collabnet start
```

Optimize PostgreSQL with vacuum

To optimize your PostgreSQL database, run a built-in utility called "vacuum."

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.)

The vacuum utility runs on a live database and, like the backup command, can be scripted to run nightly during minimal server load.

1. To vacuum the CollabNet TeamForge 5.2 database, run the `VACUUM` command as the PostgreSQL user.

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

- To set up automatic vacuuming of the database based on activity statistics, set up auto-vacuuming according to these instructions: <http://www.postgresql.org/docs/8.2/interactive/routine-vacuuming.html#AUTOVACUUM>.

Change the location of a log file

To change where log files are written to, edit the `site-options.conf` file and restart the runtime environment.

- Stop the site.

```
/etc/init.d/collabnet stop all
```

- In the `/opt/collabnet/teamforge-installer/5.2.0.0/conf/site-options.conf` file, change the value of the `LOG_DIR` variable to reflect the location where you want the log files to be written.
- Recreate the runtime environment.

```
./install.sh -V -r -d <SITE_DIR>
```


- Start the site.

```
/etc/init.d/httpd start
/etc/init.d/postgresql start
/etc/init.d/collabnet start
```

All future Apache logs, mail logs, database logs, java logs, and other logs will be written to the new location.

Back up and restore CollabNet TeamForge 5.2 data

CollabNet TeamForge 5.2 stores data in the database and on the file system. Back up all data comprehensively so that it can be restored in the event of unrecoverable failures.

-  **Note:** Please note that the items listed in this section address only the data that is either created by or a part of CollabNet TeamForge 5.2. Data that is not specific to TeamForge, such as operating system-based content, configuration files, and other third-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.

Back up CollabNet TeamForge 5.2 data

Use the `backup-data.py` utility to compress a copy of your site data to a location where you can quickly retrieve it.

This backup method requires shutting down your site briefly. If you cannot tolerate a shutdown, you might consider another backup/restore method, such as the NetApp Snapshot utility.

- Stop the CollabNet TeamForge 5.2 application server.

```
/etc/init.d/collabnet stop all
```

- Run the backup script.

```
cd <SITE_DIR>/runtime/scripts
./backup-data.py --destination=<directory name>
```

CollabNet TeamForge 5.2 creates the directory and stores the following data in it, in compressed format:

- Subversion repositories
- CVS repositories
- The data directory (/var)
- The CollabNet TeamForge 5.2 database.

3. Restart CollabNet TeamForge 5.2.

```
/etc/init.d/httpd start
/etc/init.d/postgresql start
/etc/init.d/collabnet start
```

Restore backed-up CollabNet TeamForge 5.2 data

Use the `restore-data.py` utility to bring back site data that has been backed up.

1. Stop the CollabNet TeamForge 5.2 application server.

```
/etc/init.d/collabnet stop all
```

2. Run the restore script.

```
cd <SITE_DIR>/runtime/scripts
./restore-data.py --source=<directory name>
```

where `<directory-name>` is the directory to which you backed up the data with the `backup-data.py` script. CollabNet TeamForge 5.2 unpacks the backed-up Subversion and CVS repositories, the data directory, and the CollabNet TeamForge 5.2 database.

3. Restart CollabNet TeamForge 5.2.

```
/etc/init.d/httpd start
/etc/init.d/postgresql start
/etc/init.d/collabnet start
```


Back up a PostgreSQL database


You can back up a PostgreSQL database safely while it is online by using the native `pg_dump` command.

In this example, the database is dumped into a GNU tar formatted file.

Run this command:

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

 **Note:** For this example, the name of the CollabNet TeamForge 5.2 database is assumed to be `sf`.

 **Tip:** See the PostgreSQL `pg_dump` man page for more information and examples.


Restore a PostgreSQL database


You can restore a PostgreSQL database with the native `pg_restore` command.

1. Locate the dump file you created when backing up the PostgreSQL database.
2. Shut down CollabNet TeamForge 5.2.
3. Create a database and user with the names used for CollabNet TeamForge 5.2.

4. Restore the database.

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

 **Note:** This example assumes that the name of the CollabNet TeamForge 5.2 database is `sf`.

 **Tip:** It may also be necessary to restore ownership of the restored tables to the SOURCEFORGE 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.

Appendix

A

Frequently asked questions about CollabNet TeamForge system administration

Topics:

- *What does it take to install CollabNet TeamForge 5.2?*
- *What does it mean to run CollabNet TeamForge 5.2 on a virtual machine?*
- *Why won't my CollabNet TeamForge 5.2 virtual machine installation start?*
- *Why does my CollabNet TeamForge 5.2 site show a different time than the host machine it is running on?*
- *How does CollabNet TeamForge 5.2 manage security?*
- *What is a patch?*
- *Does CollabNet TeamForge 5.2 support merge tracking?*
- *Should I move my TeamForge database to its own server?*
- *Should I move my source control application to its own server?*

What does it take to install CollabNet TeamForge 5.2?

To install CollabNet TeamForge 5.2, you download the software, make decisions about how you want the site to work, and set up data for the site to work with.

Do I need an advanced TeamForge 5.2 install?

To choose between a dedicated installation and an advanced installation, consider how your site's database and source control services will be used and maintained.

Hostname and domain name

The TeamForge installer can automatically set up your site so that users can find it at the `localhost` address. If you need to set a hostname other than `localhost`, you must edit the `HOST` variable in the site configuration file.

If you plan to have your users access your site by a URL that is different from the host name of the machine where the site is running, you will have to edit the `DOMAIN` variable in the site configuration file.

In either case, use the advanced install instructions.

Database

The database is where users' project pages, documents, tracker artifacts, tasks, discussions and other work products are stored and accessed. If you need to configure your database for your specific conditions of use, use the advanced install instructions.

Here are some reasons why you might want to customize the configuration of your site's database:

- Other applications are sharing the database instance with TeamForge.
- You plan to use an Oracle database. (The default option is PostgreSQL.)
- You plan to run your database on a separate standalone server.

Source control

Here are some reasons why you might want to customize the configuration of your site's source control service:

- You need to provide more than one Subversion server.
- You plan to run your source control service on a separate standalone server.
- You need to provide other source control services. (CVS and Perforce are supported.)

Security

- If you intend to have users access your site via SSL (using a URL that starts with `https`), you will need to edit the site configuration file. See [Protect Apache with SSL](#) on page 42 for information.
- If your site requires SELinux, you must configure your Apache service. See [Set up SELinux](#) on page 42 for information.

In either case, use the advanced install instructions.

How many servers do I need to run a CollabNet TeamForge 5.2 site?

You can run CollabNet TeamForge 5.2 on one server or split up its services among multiple servers.

CollabNet TeamForge 5.2 functionality is delivered by four applications. Each application can run on its own machine or share a machine with one or more other applications. You assign specific applications to specific machines when you configure your CollabNet TeamForge 5.2 installation by editing the `site-options.conf` file.

CollabNet TeamForge 5.2 core functionality This is known internally as the `app` server. It implements JBoss and Apache services. One and only one instance of this application must be present.

Database The `database` application handles site users' data. You set the type of database by setting the value of the `DATABASE_TYPE` token to `oracle` or `pgsql`. One and only one instance of this application must be present.

Subversion	Subversion can be used to provide source control functionality. It uses the Tomcat and Apache services. A site can have zero, one, or more than one instances of the <code>svn</code> application, running on an arbitrary number of machines.
CVS	CVS can be used to provide source control functionality. It uses the Tomcat service. A site can have zero, one, or more than one instances of the <code>cv</code> s application, running on an arbitrary number of machines.

Which application runs on which server?

You specify where each application runs by editing tokens in the `site-options.conf` file.

The applications that are to run on a server are specified as the value of the `HOST_<hostname>` token. If you assign more than one application to a host, specify a space-separated list.

Important:

- Only one host can contain the `app` application (core CollabNet TeamForge 5.2).
- Only one host can contain the `database` application.
- Only one source code server of any one type can run on a machine. You can integrate a second Subversion server if it is on a separate machine from the first Subversion server.

Here are some examples of common one-host and multi-host arrangements, as expressed in the `site-options.conf` file.

One host

In this example, all applications are running on one server named `worlddomination.supervillain.net`.

```
HOST_worlddomination.supervillain.net=app database subversion cvs
```

Two hosts

In this example, one host is running the core functionality and the database, while another host provides Subversion for source control.

```
HOST_itchy.springfield.com=app database
HOST_scratchy.springfield.com=subversion
```

Three hosts

In this example, one Subversion server runs on its own box while another Subversion server shares a machine with the core CollabNet TeamForge 5.2 application.

```
HOST_athos.musketeers.net=app subversion
HOST_aramis.musketeers.net=cvs database
HOST_porthos.musketeers.net=subversion
```

Four hosts

In this example, one instance of each application runs on its own host.

```
HOST_hearts.rummy.org=app
HOST_spades.rummy.org=database
HOST_clubs.rummy.org=subversion
HOST_diamonds.rummy.org=cvs
```

How does CollabNet TeamForge 5.2 handle third-party applications?

CollabNet TeamForge 5.2 relies on many third party applications to augment or enhance functionality. Note the limitations to support and functionality in some of these applications.

CollabNet TeamForge 5.2 integrates with additional third party applications, such as Microsoft Office 2003 and XP, and Microsoft Project 2002 and 2003. Support will always make an effort to provide assistance in using third party applications with CollabNet TeamForge 5.2. However, for complete, end-to-end support, customers should consult with the application vendor, as the vendor is best equipped to provide the depth and breadth of support necessary to use their products.

CVS

CollabNet CollabNet TeamForge 5.2 Technical Support provides best-effort support for Subversion and CVS client usage issues. CollabNet TeamForge 5.2 does not ship any source control functionality. (Neither the server daemon nor the client ships with CollabNet TeamForge 5.2.) For best results, customers should contact the vendor that supplied the source control server and client 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 32 or more CVS repositories from accessing the repositories that are alphabetically after the 31st. This is currently RedHat bug #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 CollabNet TeamForge 5.2 users who are members of 32 or more CVS repositories on a CollabNet TeamForge 5.2 CVS server.

Discussion forum threading

For CollabNet TeamForge 5.2 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. While the lack of either of the headers is not an explicit RFC violation, the inclusion of such headers is considered compliance with section 3.6.4 of [RFC 2822](#).

Microsoft Outlook and Lotus Notes are prone to sending mail without 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, mail is sent 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 therefore there are no known workarounds.

Contact your IT group or the vendor of your email client with questions or concerns.

Which ports should I keep open?


The components of a CollabNet TeamForge 5.2 installation listen on a number of operating system ports. Most of these ports are only used internally on the server for application communications. A small subset must be exposed externally to enable users to access all CollabNet TeamForge 5.2 services.

You can select your open ports in one of three ways:

- When you are installing Red Hat or CentOS, the Firewall Configuration screen lets you set up a basic firewall and allow incoming access on specific ports.
- After installation, you can launch the RedHat/CentOS Security Level Configuration Tool with the command `system-config-selinux`.
- After installation, you can edit the `/etc/sysconfig/iptables` file and specify your open ports by hand.

The following operating system level ports must be exposed. All other ports can be firewalled off to maintain security.

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). CollabNet TeamForge 5.2 discussion forums include mailing list functionality that allows users to send email to the CollabNet TeamForge 5.2 server. The James mail server included with CollabNet TeamForge 5.2 listens on port 25 to accept this mail for processing.
80 (http)	Port 80 is the default port for Web data transfer.
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 third 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).

-  **Important:** If you are running the SCM (CVS, Subversion, or Perforce) integration server on a separate physical server from the CollabNet TeamForge 5.2 standalone application server, you must expose a port on the SCM integration server on which the application server can communicate with the SCM integration server. The default is port 7080.

What does it mean to run CollabNet TeamForge 5.2 on a virtual machine?

CollabNet TeamForge 5.2 can run as a virtual machine image in a VMWare Player. You get all the functionality of CollabNet TeamForge 5.2 with the ease of installation and maintenance that comes with VMWare.

To access CollabNet TeamForge 5.2, one user (generally the site administrator) must configure and run the CollabNet TeamForge 5.2 application server in VMWare Player. When the CollabNet TeamForge 5.2 application server is running, other users can access it via a Web browser. These users do not need to run VMWare Player.

The CollabNet TeamForge 5.2 download may also run on some other VMWare products, such as VMWare Workstation 5.5. However, these instructions are only for using VMWare Player.

Why won't my CollabNet TeamForge 5.2 virtual machine installation start?

CollabNet TeamForge 5.2 won't start, or you receive an error message when trying to access your site.

You may be encountering one of the following issues:

- The CollabNet TeamForge 5.2 application server is not running.
- Your organization has exceeded your maximum number of licensed users for the CollabNet TeamForge 5.2 download.
 - The free trial version supports up to 3 users at no charge.
 - The Team edition supports up to 25 users.

To purchase additional licenses, visit <http://www.collab.net/products/sfee/buyit.html>


- You are attempting to run CollabNet TeamForge 5.2 on an unsupported VMWare product. The following legacy VMWare product versions are not supported:
 - VMWare ESX Server 2.x
 - VMWare GSX Server 3.x
 - VMWare ACE 1.x
 - VMWare Workstation 4.x

For more information about VMWare Player and similar products, see <http://www.vmware.com/products/player/>

Why does my CollabNet TeamForge 5.2 site show a different time than the host machine it is running on?


In some cases it is possible for the clock in the CollabNet TeamForge 5.2 VMWare image to drift from that of the host machine. If you notice this issue, you can set the CollabNet TeamForge 5.2 VMWare image to synchronize time with an external NTP server.

A script is provided to enable you to configure time synchronization easily. The `configure-ntp.sh` script sets up a manual periodic time sync once per hour between the VMWare image and the NTP server.

-  **Important:** Before running this script, your virtual machine must be able to access an external NTP server. If your virtual machine is running inside a firewall, and is unable to access an external public NTP server, you may need to talk to your system administrator to find an accessible NTP server within your network.

While logged into the virtual machine, run `/root/configure-ntp.sh <ntp server>`.

If you do not enter an NTP server, the script will try to use `pool.ntp.org`, a publicly available time service, by default.

 **Note:** VMware advises against setting up the VMware image to use NTP directly because it can interfere with VMware's own built-in time syncing mechanism.

For detailed information about timekeeping in VMware, see http://www.vmware.com/pdf/vmware_timekeeping.pdf

How does CollabNet TeamForge 5.2 manage security?

CollabNet TeamForge 5.2 is a secure, centralized, enterprise-grade solution for optimizing distributed development. A number of factors go into ensuring security.

How does CollabNet TeamForge 5.2 help protect data access?

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 CollabNet TeamForge 5.2 installations must be secured with a firewall restricting the access to specific web server ports. Neither the CollabNet TeamForge 5.2 application server nor the backend servers should ever be exposed to the Internet.

The CollabNet TeamForge 5.2 application and the backend servers can be further secured by limiting their access within the private network (Intranet). For CollabNet TeamForge 5.2 application to function effectively, the following conditions must be met.

- 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 CollabNet TeamForge 5.2 application server.
 - Send mail to CollabNet TeamForge 5.2 mail server via SMTP (port 25).
 - The SCM server through a secure protocol such as SSH (port 22).
- The web server must have access to the application server (typically port 8080).

 **Note:** 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 CollabNet TeamForge 5.2 for commit notifications.
- The mail server must be able to deliver messages across the firewall.

Authentication and authorization

To secure sensitive data, CollabNet TeamForge 5.2 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.

CollabNet TeamForge 5.2 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 CollabNet TeamForge 5.2.

What user activities are tracked?

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.


CollabNet TeamForge 5.2 application also audits every change made to application objects (trackers, artifacts, documents, etc.) within the system. Administrative actions such as adding a new user and 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. CollabNet TeamForge 5.2 auditing tools are a powerful way to track unwanted and/or unauthorized changes within the system.

How does CollabNet TeamForge 5.2 help protect my 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).

 **Note:** The web server employed by a CollabNet TeamForge 5.2 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 CollabNet TeamForge 5.2, sensitive data includes user passwords. CollabNet TeamForge 5.2 encrypts no other data. Since user passwords are used for authentication purpose only, CollabNet TeamForge 5.2 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.

Does CollabNet TeamForge 5.2 work with LDAP?

You can have your CollabNet TeamForge 5.2 installation authenticate against your corporate LDAP server.

When you use LDAP authentication, users present their credentials to a central authentication server rather than signing into CollabNet TeamForge 5.2 directly. This is handy when users want to use a variety of different resources without having to sign onto each one separately.

LDAP authentication is optional. You can use either CollabNet TeamForge 5.2 authentication or LDAP authentication, but you cannot use both with a single CollabNet TeamForge 5.2 installation.

When CollabNet TeamForge 5.2 is configured to authenticate against an LDAP server and the LDAP server is down, all CollabNet TeamForge 5.2 authentication is disabled until the LDAP server is restored.

If a user does not exist on the LDAP server, or is deleted from the server, that user cannot log into CollabNet TeamForge 5.2. The admin user is the only exception to this rule.

SCM authentication is not managed by LDAP, but each CollabNet TeamForge 5.2 user's SCM password is synchronized automatically with his or her LDAP password upon logging into CollabNet TeamForge 5.2.

If your CollabNet TeamForge 5.2 installation has user accounts that were created before LDAP integration was enabled, these user accounts remain active. However, their usernames in CollabNet TeamForge 5.2 must exactly match their usernames on the LDAP server. If this is the case, these users can log into CollabNet TeamForge 5.2 using their LDAP passwords.

If a password is used in LDAP that is shorter than the minimum allowable password length in CollabNet TeamForge 5.2, you cannot create the user in CollabNet TeamForge 5.2.

J2EE Architecture and security

CollabNet TeamForge 5.2 is a J2EE application that employs three-tier architecture to provide a secure environment for mission-critical data.

In a multi-tier architecture, access to each tier is restricted to the tier above it, effectively securing the tiers behind the firewall. 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.

What security tools come with CollabNet TeamForge 5.2?

In addition to employing industry standard security protocols, CollabNet TeamForge 5.2 provides an extensive access control model for fine-grained control and powerful tools to audit and track changes.

 **Note:** Although CollabNet intends CollabNet TeamForge 5.2 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. CollabNet TeamForge 5.2 can 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.

Cookies

CollabNet TeamForge 5.2 requires browsers to support cookies. Cookies are used for the sole purpose of managing user sessions. CollabNet TeamForge 5.2 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

CollabNet TeamForge 5.2 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 says:

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 CollabNet TeamForge 5.2 authenticates a user's login information. A session is invalidated when one of following events occur:

- The user explicitly logs out of CollabNet TeamForge 5.2 .
- When the user's session times out.

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

Passwords

CollabNet TeamForge 5.2 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 CollabNet TeamForge 5.2 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, CollabNet TeamForge 5.2 does not perform any kind of password strength checking (e.g. CollabNet TeamForge 5.2 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

CollabNet TeamForge 5.2 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.

What is a CERT advisory?

CollabNet Product Support monitors the CERT coordination center (<http://www.cert.org/>) for notification of vulnerabilities or exploits against applications that CollabNet TeamForge 5.2 provides.

If CollabNet Technical Support identifies an advisory that may indicate potential challenges for users who have deployed CollabNet TeamForge 5.2, Support proactively releases a notification and a statement of action. CollabNet will provide product updates as it deems appropriate or necessary.

What is a patch?

A patch is a package of code that fixes or adds to the functionality of a CollabNet product. Patches are also known as "component upgrades."


Things to know about patches

- Patches are cumulative. You don't need to apply multiple patches sequentially to get to the desired patch level. You can move up (or down) one or more patch levels with a single operation.
- The Level option (-l) allows you to downgrade or upgrade to any patch level (within the maximum available in the cumulative patch).
- The Rollback option (-r) allows you to revert the site to the previous patch level it was at, before the current patch was applied.
- The Uninstall option (-u) allows you to downgrade the patch level on the site by one.
- When a patch installation fails you can use the Force option (-F) to proceed, without manually uninstalling previous patches.
- The system displays a summary of what happens during the patch installation.
- Before proceeding with the patch installation, you can use the "dry run" mode (-t option) to see the summary of actions that will be performed during the installation.

Best practices

Before applying a patch, note the following principles.

- The upgrade scripts are usable only with an existing installation.
- No data migration will occur if any changes have been made to the database schema.
- You must use the `sudo` command or have an account that is equivalent to root in order to complete a patch installation successfully.

 **Important:** Before installing a patch, verify that it has been fully tested and qualified.

Does CollabNet TeamForge 5.2 support merge tracking?

The Subversion repositories that are installed with CollabNet TeamForge 5.2 run on Subversion 1.5, which supports merge tracking.


Any Subversion 1.5 servers you have integrated with CollabNet TeamForge 5.2 support merge tracking. If you need the merge tracking feature and your Subversion server is running a version earlier than Subversion 1.5, you must upgrade to Subversion 1.5 to get this functionality.

If you used `svnmerge.py` (<http://www.orcaware.com/svn/wiki/Svnmerge.py>) to do merge tracking before Subversion 1.5, and you want to convert your `svnmerge.py` data to the Subversion 1.5 merge tracking data format, CollabNet provides a migration tool here:


<http://svn.collab.net/repos/svn/trunk/contrib/client-side/svnmerge/svnmerge-migrate-history.py>

Should I move my TeamForge database to its own server?

If you expect your site to have heavy user traffic, you may want to run the site's database on its own server.

 **Note:** Before moving your database to its own server, make sure you have access to someone with advanced skills in the database service you are using.

The advantage of hosting a service on a separate server is that it does not share CPU, RAM or I/O bandwidth with the server that is hosting the main TeamForge application.

 **Note:** Each TeamForge site can have only one database server.

To help decide whether you need a separate database server, consider these approximate values:


	Shared TeamForge-database server	Standalone database server
Daily users	Fewer than 1000	More than 1000
Daily discussion forum entries	Fewer than 1000	More than 1000

Should I move my source control application to its own server?

If you anticipate heavy source code check-in and check-out traffic, consider setting up the source code application on its own server.

To host your source control services on their own server, you must set up a source code repository server and integrate it with TeamForge. You can integrate any number of source code servers with your TeamForge site.

The advantage of hosting a service on a separate server is that it does not share CPU, RAM or I/O bandwidth with the server that is hosting the main TeamForge application.

 **Note:** If you need to move a source code integration, contact your CollabNet representative for help.

To help decide whether you need a separate source control server, consider these approximate values:

	Shared TeamForge-SCM server	Standalone SCM server
Daily source code commits	Fewer than 1000	More than 1000

Appendix

B

Reference information for CollabNet TeamForge system administration

Topics:

- [Install reference](#)
 - [Scripts](#)
 - [Log files](#)
 - [Configuration files](#)
-

Install reference

This reference information includes details to support installing the application.


Minimum hardware requirements for CollabNet TeamForge 5.2

The following hardware is the minimum recommended for the server on which CollabNet TeamForge 5.2 is installed and the server on which the database is installed.

CollabNet TeamForge 5.2 requires the following minimum server characteristics.

- 2 x CPU 2GHz
- 2 GB RAM
- >40GB hard drive

Required hard drive capacity depends on the estimated amount of document and file release uploads.

 **Note:** We strongly recommend running the CollabNet TeamForge 5.2 application and its database on separate physical servers. Each server should meet the same dual-processor, 2-GHz standard.

Supported software for CollabNet TeamForge 5.2

This is the official list of software that is compatible with CollabNet TeamForge 5.2.

Operating systems

CollabNet TeamForge 5.2 has been tested on these operating systems:

Operating system	Version	Architecture	Restrictions
Red Hat Enterprise Linux	5.3	32-bit	Must have RHN or equivalent
Red Hat Enterprise Linux	4.7	32-bit	
Red Hat Enterprise Linux	4.7	64-bit	
CentOS	5.2	32-bit	
CentOS	4.7	32-bit	
CentOS	4.7	64-bit	

 **Important:** Red Hat Enterprise Linux 5.3 machines must have access to the Red Hat Network or equivalent (satellite server, spacewalk, or RHN proxy). For more information, see www.redhat.com.

Databases

These database products have been tested with CollabNet TeamForge 5.2:

- PostgreSQL 8.2.12
- Oracle 10.2

Browsers

These browsers have been tested with CollabNet TeamForge 5.2:

- Mozilla Firefox 3.0.x
- Microsoft Internet Explorer 6.0.x
- Microsoft Internet Explorer 7.0.x

Software configuration management tools

These software configuration management (SCM) tools have been tested with CollabNet TeamForge 5.2:


- CVS 1.11.x
- Subversion 1.5.5 (FSFS)
- Perforce 2006.2

Java SDK

Java SDK 1.5.0_12 has been tested with CollabNet TeamForge 5.2.

Oracle client

Oracle 10gR2 - 10.2.0.1.0 Standard Edition has been tested with CollabNet TeamForge 5.2.

 **Note:** The Express Edition is not supported.

Microsoft applications

These Microsoft applications have been tested with CollabNet TeamForge 5.2:

- Microsoft Project 2002 (with Service Pack 1) on WinXP Service Pack 2 and Win20002 Service Pack 4
- Microsoft Project 2003 (with Service Pack 1) on WinXP Service Pack 2 and Win20002 Service Pack 4
- Microsoft Office XP (with Service Pack 3) on WinXP Service Pack 2 and Win20002 Service Pack 4
- Microsoft Office 2003 (with Service Pack 1) on WinXP Service Pack 2 and Win20002 Service Pack 4

LDAP (Lightweight Directory Access Protocol) application

OpenLDAP 2.3.27-5 has been tested with CollabNet TeamForge 5.2.

Packages required for 32-bit Red Hat 5

CollabNet TeamForge 5.2 requires these packages when running on the 32-bit architecture of Red Hat Enterprise Linux 5.3

Packages are listed in the order in which they are installed.

Package	Shipped with OS	Required for TeamForge 5.2	Tolerance	Compiled by CollabNet
Python-curl	N/A	7.15.5-1.2	Should be compatible with the python installed.	No
httpd	2.2.3	2.2.3	2.2.3-*. Should be compatible with apr installed	No
Jdk	N/A	1.5.0_12	>= 1.5	No
Postgres	8.1.4	8.2.12	8.3.*	No
Neon	0.25.5	0.27.2	>=0.27.2	No
Subversion	1.4.2	1.5.5	1.6.*	No
Mod_dav_svn	N/A	1.5.4	Should be compatible with the subversion installed.	No
CVS	1.11.22-5	1.11.22-5	1.11.22-5	No
Python-Chardet	N/A	1.0-1	1.0-1	Yes
ZSI	N/A	1.7-2	1.7-2	Yes
Python-fpconst	N/A	0.7.2-3	0.7.2-3	Yes

Package	Shipped with OS	Required for TeamForge 5.2	Tolerance	Compiled by CollabNet
SOAPpy	N/A	0.11.6-3	0.11.6-3	Yes
apr	1.2.7	1.2.7	>= 1.2.7	No
Apr-util	1.2.7	1.2.7	Should be compatible with the apr installed.	No

Packages required for 32-bit Red Hat 4

CollabNet TeamForge 5.2 requires these packages when running on the 32-bit architecture of Red Hat Enterprise Linux 4.7

Packages are listed in the order in which they are installed.

Package	Shipped with OS	Required for TeamForge 5.2	Tolerance	Compiled by CollabNet
Python-curl	N/A	7.12.1-1.2	Should be compatible with the python installed.	No
httpd	2.0.52	>=2.0.52	2.0.52-*. Should be compatible with apr installed	No
Jdk	N/A	1.5.0_12	>= 1.5	No
Postgres	7.4.13	8.2.12	8.3.*	No
Neon	0.24.7	0.27.2	>=0.27.2	No
Subversion	1.1.4	1.5.5	1.6.*	No
Mod_dav_svn	1.1.4	1.5.4	Should be compatible with the subversion installed.	No
CVS	1.11.17	1.11.17	>=1.11.17	No
Python-Chardet	N/A	1.0-1	1.0-1	Yes
ZSI	N/A	1.7-2	1.7-2	Yes
Python-fpconst	N/A	0.7.2-3	0.7.2-3	Yes
SOAPpy	N/A	0.11.6-3	0.11.6-3	Yes
apr	0.9.4	0.9.12	>= 0.9.12	No
Apr-util	0.9.4	0.9.12	Should be compatible with the apr installed.	No

Packages required for 64-bit Red Hat 4

CollabNet TeamForge 5.2 requires these packages when running on the 64-bit architecture of Red Hat Enterprise Linux 4.7

Packages are listed in the order in which they are installed.


Package	Shipped with OS	Required for TeamForge 5.2	Tolerance	Compiled by CollabNet
Python-curl	N/A	7.12.1-1.2	Should be compatible with the python installed.	No

httpd	2.0.52	>=2.0.52	2.0.52-*. Should be compatible with apr installed	No
Jdk	N/A	1.5.0_12	>= 1.5	No
Postgres	7.4.13	8.2.12	8.3.*	No
Neon	0.24.7	0.27.2	>=0.27.2	No
Subversion	1.1.4	1.5.5	1.6.*	No
Mod_dav_svn	1.1.4	1.5.4	Should be compatible with the subversion installed.	No
CVS	1.11.17	1.11.17	>=1.11.17	No
Python-Chardet	N/A	1.0-1	1.0-1. Should be compatible with the python installed	Yes
ZSI	N/A	1.7-1	1.7-1	Yes
Python-fpconst	N/A	0.7.2-3	0.7.2-3. Should be compatible with the python installed	Yes
SOAPpy	N/A	0.11.6-3	0.11.6-3. Depends on python-fpconst	Yes
apr	0.9.4	0.9.12	>= 0.9.12	No
Apr-util	0.9.4	0.9.12	Should be compatible with the apr installed.	No

Hardware and software requirements for the CollabNet TeamForge 5.2 download

To run the CollabNet TeamForge 5.2 system must meet some minimal requirements.

- An operating system that VMware Player can run on. For Windows and Linux versions supported by VMware Player, see the [VMware Player documentation](#).
- 2GB system RAM
- 8GB available disk space
- 2 Ghz Pentium 4 or equivalent processor

 **Note:** CollabNet TeamForge 5.2 may also run on some other VMware products, such as VMware Workstation 5.5. However, these instructions are only for using VMware Player.

Contents of the CollabNet TeamForge 5.2 download package

The CollabNet TeamForge 5.2 download is provided in a .zip file that also contains VMware Player and associated installation and configuration instructions.

 **Note:** The .zip file is available from <http://www.collab.net/products/sfee/tryit.html>. Unzip it to extract the files.

The .zip file contains:

README.txt	Quick start instructions.
image_files	The CollabNet TeamForge 5.2 download VMware image.
VMwarePlayer	The installers and user documentation for the VMware Player. Versions for Windows and Linux are included.


Scripts

System administrators can use these utilities to control the behavior of the application.

bootstrap-data.sh

The `bootstrap-data.sh` script prepares application and database data for new installations. Preparing application and database data is referred to as "bootstrapping" the data.

Overview

 **Important:** This script is only for new installations. If you run it on a site that already has data, all data will be wiped.

This script resides in the `<installation_source>` directory and calls the `wrapper-bootstrap-data.py` script when run. The `[log_file_directory]/runtime/bootstrap.log` file is created when this script is run. All success and error messages from this script are written to this log file.

Usage

Run this script as follows:

```
cd <installation_source>

./bootstrap-data.sh -d <SITE_DIR>
```

Example

The following command forces a bootstrap of the data, showing all actions on the screen:

```
./bootstrap-data.sh -n -F -V -d <SITE_DIR>
```

Options

The following options are available for the `bootstrap-data.sh` script:

-d --directory	Specify installation directory. This argument is required.
-h --help	Provides a list of all available options for this script.
-n --non-interactive	Runs the script in a non-interactive mode. The script will fail with an error message when used with this option if an existing <code>[DATA_DIR]</code> is located. You can use the <code>-F</code> option to force bootstrapping on sites that have an existing <code>[DATA_DIR]</code> .
-F --force	This option is only valid when the <code>-n</code> option is used. This option forces the bootstrapping of data when a <code>[DATA_DIR]</code> exists.
-V --verbose	Writes all script actions to the screen. Without this option the script runs silently and logs messages to the <code>[log_file_directory]/runtime/bootstrap.log</code> file.
-q --quiet	Do not show script output.
-f --site-options=[filename]	Points to the <code>site-options.conf</code> configuration file for the site. This argument is optional.

Cluster location


This script runs on the application server machine.

The collabnet script

Run this script to start or stop TeamForge, or to get the status of the application or a component.

Overview

You can use this script to start or stop the application as a whole or to start and stop an individual service. You can also use it to determine the status of an individual service.

 **Important:** On production sites, this script must be invoked by the root user.

Usage

Run this script as follows:

```
<SITE_DIR>/runtime/scripts/collabnet [--verbose|-V] [--service|-s serviceName]
<command>
```

For example, the following command checks the status of the jboss component:

```
<SITE_DIR>/runtime/scripts/collabnet -s jboss status
```

Parameters

Command	Action to perform. The supported commands are:
start	Starts the application / service
stop	Stops the application / service
status	Provides the status of the service(s)
restart	Restarts the application / service
help	Prints this message and exits.
-s service serviceName	Perform the command for the service serviceName
verbose	Print debug messages
-q --quiet	
-F --force	Force option to perform the specified operation forcefully
-S --silent	To perform the operation silently without providing the output

Logging

collabnet writes entries to the following logs:


- `log/runtime/service.log`: The master service log.
- `log/{service}/service.log`: Log entries from starting up individual services end up in the `service.log` file of the corresponding service log folder (e.g. `log/apps/service.log`)

environment_check.sh

The `environment_check.sh` script verifies whether all environment packages required for installing CollabNet TeamForge are present.

Overview

Use the `environment_check.sh` script to verify that you have all required environment packages for installing SourceForge.

 **Note:** You may be prompted to manually remove any older packages identified on your system before running the `install.sh` script.

Usage

From the <INSTALLATION_SOURCE> directory, run this script as follows:

```
./environment_check.sh
```

install.sh

The `install.sh` script handles all operations related to installing and removing the application.

Options**-r | --runtime**

Create the runtime instance using the site configuration file

-R | --internalruntime

Create the internal runtime instance using the site configuration file

-F | --force

Force the operation that is performed wherever appropriate (eg. install/uninstall)

-i | --install

Install application packages

-u | --uninstall

Uninstall application packages

-S | --startnow

Start application services after completing the other operations (if possible)

-s | --startup

Start application services on reboot and start it now: equivalent to `-I -S`

-d | --directory type='str', argname='installation_base_dir',


Path where the site would get installed

-f | --siteoptionsfile

Path to the site configuration file (default: `./conf/site-options.conf`)

-c | --conf-file

The environment configuration file (usually of the form `environment-<platform>.conf`), used to identify the platform.

 **Note:** This is not to be confused with the `site-options.conf` file.

-I | --initscript

Start application services on reboot

-V | --verbose

Show all output in noninteractive mode

-E | --check_environment

Check if the system environment is suitable for this installation

-a | --all

Performs these operations in sequence - Install, create runtime and setup initscripts (equivalent to: `-I -r -i`)

-D | --debug

Displays traceback errors if any

-C | --cleanup

Stop/Kill the application processes, wipes out application packages and site directory.



Caution: THIS OPTION WILL WIPE OUT THE SITE DATA.

-n | --noninteractive

Used to run the installer in non-interactive mode

pbl.py

The `pbl.py` utility enables you to upload files to the Project Build Library and perform various operations on them.

Options

--help -h	Print out a help message and exit.
--api-user -u username	Your TeamForge Lab Management login name. Required for all upload operations.
--api-key -k key	Your TeamForge Lab Management API key. Required for all upload operations.
--api-url -l url	The URL to the TeamForge Lab Management API's. Will generally be <code>https://\$external_host/cubit_api/1</code> . Required for all upload operations.
--comment -c "your comment"	Print out a comment on this operation. The comment is always optional. The comment string will be logged in the audit log, but is not recorded in the PBL. For example, if you are deleting some files, you might want to use a comment to explain why you were deleting those files, for future auditing purposes.
--verbose -v	Print out more detail on what the <code>pbl.py</code> is doing.
--xml-server-output --xml	If this option is not specified, the <code>pbl.py</code> client reads in the XML returned from the server and presents the results to you in nicely formatted text. If you'd like to instead see the raw XML returned from the server, select this option.
--no-auth-cache	As a convenience, the <code>pbl.py</code> function caches the value of the <code>--api-user</code> and <code>--api-key</code> parameters in your home directory, in a subdirectory named <code>.TeamForge Lab Management</code> , the first time a successful authentication is performed against the server. This is analogous to the Subversion client's use of the <code>.subversion</code> directory to store authentication credentials. Selecting the <code>--no-auth-cache</code> option turns off this caching.
--project p projname	The TeamForge Lab Management project in which the file you are operating on is located.
--type t {pub priv}	The visibility type of the file, either <code>pub</code> (the file is in the public area of the PBL) or <code>priv</code> (the file is in the private area of the PBL).
--remotepath r path	The remote path on the server, excluding the base directory, the project, and the visibility type. Examples are below.

snapshot.py

Use this script as a debugging tool to troubleshoot system errors. It records a snapshot of the current state of the machine.

Overview

Run this script manually to generate debugging information before restarting the instance.

Usage

Run this script as follows:

```
<SITE_DIR>/runtime/scripts/snapshot.py
```


Options

The following options are available for the `snapshot.py` script:


- h|--help** Provides information on using the script.
- extra** An arbitrary command whose output should be placed in the generated log file. For example, you can have `snapshot.py` execute the `lsof` command like this:

```
--extra '/usr/sbin/lsof -n -P -b -i -U'
```

Enclose commands with options in quotes.
- v|--verbose** Provides output on the actions performed by the script.

 **Note:** The output from `snapshot.py` is written to a log file in the `[LOG_DIR]/runtime` directory. Use the output (`snapshot.log`) to troubleshoot any system or CollabNet related errors.

Cluster location

 **Important:** The `snapshot.py` script generates a log file for the node on which it is run. When a CollabNet site is deployed on a cluster and you need information to troubleshoot problems, it is recommended that you run this script on all the nodes.

upgrade-site.sh

With this script, you can perform a cumulative patch upgrade or downgrade on a running instance.

Overview

This is a wrapper for the `upgrade.py` script.

The script verifies the following:

- The user invoking the script is the equivalent of a root user.
- The specified directory has a valid SourceForge installation.

It performs the following actions depending on the options specified:

- Displays a summary of what would happen during the patch installation.
- Downgrades or upgrades the site to the specified patch level.
- Reverts the site to the previous patch level it was at, before the current patch was applied.
- Downgrades the patch level on the site by one.
- Starts SourceForge after successfully installing the patch.
- Allows a test "dry run" of the patch installation.

Usage

Run this script as follows:



```
./upgrade-site.sh -d <INSTALL_DIR> [-r] [-u] [-t] [-l level] [-f file] [-n]
[-h] [-V] [v]
```


Example

To perform a component upgrade from a base SourceForge installation (patch level 0) to patch level 2, use this command:

```
sudo ./upgrade-site.sh -t -d /opt/collabnet/teamforge -l 2
```

Options

-f [manifest] --file [manifest]	The manifest file with the appropriate information for this upgrade.
-d [INSTALLATION_DIR] --directory [INSTALLATION_DIR]	The directory where the application is installed.
	 Note: This option is required.
-r --rollback	Rolls back the previous (most recently applied) patch. For example, if you upgrade the site from patch level 1 to patch level 4, and then run <code>upgrade-site.sh</code> with this option, the resulting patch level on the site is patch 1.
-l [level] --level [level]	The patch level to which the SourceForge site must be upgraded (or downgraded).
-V --verbose	Displays script output including traceback errors. If this option is not used, the script displays error messages but not the actual traceback errors.
-v --version	Displays the script version.
-n --noninteractive	Non-interactive mode.
-t --testrun	Displays a summary of the actions that will be performed as part of the upgrade or downgrade. Use this option to view a description of what would take place during a patch upgrade (or downgrade) before you actually apply the patch.
	 Note: You must use this option along with the <code>l</code> , <code>r</code> , <code>u</code> , or <code>f</code> options.
-u --uninstall	Decrements the patch level on the site by one. For example, if you upgrade the site from patch level 1 to patch level 4, and then run <code>upgrade-site.sh</code> with this option, the resulting patch level on the site is patch 3.
-h --help	Prints usage information.

 **Note:** Do not use the following combinations of options in the same command:

- `-u` (uninstall) with `-r` (rollback)
- `-f` (manifest) with `-l` (level)
- any combination of `-u`, `-r`, `-l`, `-f`

If you do, the script exits with a corresponding error message.

Log files

System administrators can use logs to debug problems and ensure that the application is performing to expectations.

JBoss logs

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

boot.log	Logs the JBoss startup and shut down notifications. This log is overwritten each time JBoss is (re)started.
localhost_access	The Records access to the application from a remote host, similar to the Apache <code>access_log</code> . This log is rotated each day, and the files have a date stamp appended to their name, such as <code>localhost_access2004-11-26.log</code> .
server.log	Logs all the activities of the application server, including any exceptions. This log is the best place to begin debugging CollabNet TeamForge 5.2 server error exception ids (exid).
session-info.log	Records when new sessions are created. This log is overwritten each time JBoss is (re)started.

vamessages.log	Records CollabNet TeamForge 5.2-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 <code>vamessages.log.1</code> and <code>vamessages.log.2</code> .
-----------------------	---

Oracle logging

The most important Oracle log is the `alert` log, which is found in `$ORACLE_HOME/admin/$SID/bdump/alert_$SID.log`.

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/`. Many logs are stored under this directory hierarchy, but `alert` is the most important. 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 more information, as well as support in the maintenance of an Oracle database, contact Oracle Support or Oracle's [Metalink](#) site.

SCM (CVS, Subversion, and Perforce) logs

Software configuration management (SCM) servers generate several logs from the CollabNet TeamForge 5.2-managed integration server in `<install_dir>/log`. Not all of them are unique to CollabNet TeamForge 5.2; 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.
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 <code><install_dir>/log</code> .

Email logs

Both the CollabNet TeamForge 5.2 email and search backends are managed from a parent daemon known as Phoenix. If the mail backend is not operating properly, the first troubleshooting step is to check the `phoenix.log` to see if it encountered difficulties starting up.

Overview

The Phoenix daemon logs its activities to the `phoenix.log` file, which is stored under `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 CollabNet TeamForge 5.2 standalone server init script.

CollabNet TeamForge 5.2 email is handled by the JAMES server. JAMES logs all of its activities under `install_dir/james/james-2.2.0/apps/james/logs`. 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`).

Active logs

Sixteen different logs are created by `james` for different components of its functionality. This topic describes only the ones that are used actively by CollabNet TeamForge 5.2.

<code>james-$\\$date.log</code>	The James log records the overall mail handling behavior of the James server.
<code>mailet-$\\$date.log</code>	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.
<code>mailstore-$\\$date.log</code>	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.
<code>smtpserver-$\\$date.log</code>	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.
<code>spoolmanager-$\\$date.log</code>	The spoolmanager log records the processing of mail spools. This log could be of value in troubleshooting mail delivery or handling problems.

Search logs

Both the CollabNet TeamForge 5.2 search and email backends are managed from a parent daemon known as Phoenix. If the search backend is not operating properly, the first troubleshooting step is to check the `phoenix.log` file to see if it encountered difficulties starting up.

The Phoenix daemon logs its activities to the `phoenix.log` file, which is stored under `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 CollabNet TeamForge 5.2 standalone server init script.

Once started successfully, the search server waits for new content to be indexed or searches to be performed. The search server logs its activities under `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.


Project Build Library audit log

You can use this screen to view the complete list of actions performed in the Project Build Library.

Contents

Information about the following types of actions is displayed in this screen:

- Change a File Description
- Create a Directory
- Delete a File or Directory
- Download a File
- Move a File or Directory
- Upload a File

 **Note:** The value displayed in the **Event** field is the value passed in the `--comment` parameter from the Project Build Library client.

Getting there

On the project home page, click **Build Library** in the left navigation bar and select the **Audit Log** tab.

Access

This screen is accessible for all users who have at least the view permission for the project.

Profile audit log

You can use this screen to view the complete list of actions performed on a profile.

Getting there

On the **Profile Library** screen, click the **Audit Log** tab.

Access

This screen is accessible for all users who have at least the view permission for the project to which the profile is allowed.

x

Example

When a user updates any of the profile fields on the **Profile Admin** screen, the following details are displayed in this screen:

- The old value for the field.
- The new value for the field.
- The name of user who updated the field.
- The time when the change occurred.

User Audit Log

You can use this screen to view the list of actions performed by the user in the TeamForge Lab Management system.

For example, when a user logs into the web interface of the TeamForge Lab Management system, the event is displayed in this screen.

Access

This screen is accessible to all users who have at least the Domain Administrator role.

Getting there

On the **Administration** tab, click **User Audit Logs** in the left navigation bar.

Host audit log

You can use this screen to view the complete list of actions performed on a host.

Getting there

On the TeamForge Lab Management Host home page, click the **Audit log** tab.

Access

This screen is accessible for all users who have at least the view permission for the project to which the host is assigned.

Example

When the IP address for the host is changed, the following details are displayed in this screen:

- The old IP address.
- The new IP address.
- The name of user who changed the IP address.

- The time when the change occurred.

Project audit log

The **Project audit log** screen shows the complete list of changes applied to a project.

Getting there

On the TeamForge Lab Management Project home page, click **Audit Logs** in the left navigation bar.

Access

This screen is accessible for all users who have at least the view permission for the project.

Example

When a profile is added to the list of buildable profiles for this project, the following information appears on this screen:

- The action that was taken.
- The user who performed the change.
- The time when this change occurred.

Configuration files


Edit these configuration files to get the behavior you want.

The patch manifest file

The patch manifest file contains all the information about the patch.

Overview

The manifest file for each patch is named `manifest-[patch#]`. The manifest file is a text file containing a set of configuration tokens.


 **Note:** The first patch is named `manifest-1`.

Contents

The manifest file contains these tokens:

PATCH_LEVEL

The patch level which this patch provides.

 **Note:** The `PATCH_LEVEL` value is used (along with information in `[DISTRIBUTION_DIR]/version/core-version.txt`) to fill in `[DISTRIBUTION_DIR]/conf/patches` with the current release and patch level. If `[DISTRIBUTION_DIR]/conf/patches` does not exist, it is created.

PATCH_DESCRIPTION


A description of the patch.

UNINSTALL_LIST

A list of RPMs to uninstall (using relative paths, comma separated).

INSTALL_LIST

A list of RPMs to install (using relative paths, comma separated).

 **Note:** Comments in the manifest file are identified by a leading hash (#).

login-config.xml

This is the sample application-policy block that you can copy into your login-config.xml file to support LDAP authentication.

Notes

Replace the default application-policy block of the login-config.xml file with this code, then make the modifications specified in [Set up LDAP integration for the CollabNet TeamForge 5.2 server](#) on page 32. Option values that must be modified are highlighted in bold.

- 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 `<principalDNPrefix><username><principalDNSuffix>`.
- In this example application-policy block, the username is stored in the People organizational unit in the dev.sf.net domain. This is represented as `,ou=People,dc=dev,dc=sf,dc=net`
- 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.

Sample code

```
<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>
```

httpd.conf

These are the changes you must make 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 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
RewriteRule ^/integration/(.*) http://localhost:7080/integration/$1 [P]
ProxyPassReverse / http://localhost:8080/
ProxyPassReverse / http://localhost:7080/
##
# end SFEE configuration
##

```

iptables

This is the `/etc/sysconfig/iptables` output that will enforce the recommended security configuration.

```

# 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
```