Administrating TeamForge 6.1.1 on Red Hat

Contents

Red Hat system administrator how-tos

A system administrator provides the infrastructure that lets site administrators, project managers and members collaborate effectively.

Install CollabNet TeamForge 6.1.1 on Red Hat

When you finish this sequence of tasks, you will have a working TeamForge 6.1.1 site tailored to the specific requirements of your user base and environment.

Installing TeamForge is not difficult, but it can be complex. To succeed, you should be familiar with essential Linux commands and terminology. Most commands are given explicitly, but you will sometimes have to substitute values corresponding to your own specific environment.

☞ **Note:** For the hardware and software required to run TeamForge, see *Hardware and software requirements for CollabNet TeamForge on a virtual machine*.

Plan your CollabNet TeamForge 6.1.1 installation

Before you install TeamForge 6.1.1, let's take a look at the product from a system administrator's perspective, so that you know exactly what you are getting into.

Overview

A TeamForge site consists of a core TeamForge application and several tightly integrated services that support it.

- The core TeamForge application provides the Web interface that users see, and the API that other applications can interact with. It also includes the file system where some user content is stored, such as wiki pages.
- The site database is where most of the user-created content is stored and accessed. Documents, discussion posts, tracker artifacts, project administration settings: all that sort of thing lives in the database.
- The source control server ties any number of Subversion, CVS or Perforce repositories into the TeamForge site. These links are called "integrations" because they provide rich two-way coupling, not just a link, between the features of the source control tool and the features of the TeamForge site.
- The reporting server pulls data from the site database to populate graphs and charts about how people are using the site.
- The datamart is an abstraction of the site database, optimized to support the reporting functionality.

Install sequence

TeamForge supports multiple options for customizing and expanding your site to fit your organization's unique use patterns.

In the default setup, all services run on the same box as the main TeamForge application. But in practice, only the TeamForge application *needs* to run on the TeamForge application box. The other services can share that box or run on other boxes, in almost any combination. When you spread your services around to multiple boxes, you must do some configuration to handle communication among the services.

You should assess your own site's particular use patterns and resources to decide how to distribute your services, if at all. For example, if you anticipate heavy use of your site, you will want to consider running the site database, the source control service, or the reporting engine on separate hardware to help balance the load.

CollabNet has extensively tested six typical configurations:

- *Out of the box*: Everything runs on one machine. Perfect for sites with fewer than 100 users.

- *Coder's playground*: Everything runs on one box except source control, which has its own box.
- *Project office*: Everything runs on one box except the reporting data. Site administrators can run frequent reports without overloading the rest of the site.
- *Project office, high volume*: For the larger site. Source control and the reporting data get their own server.
- *DBA's choice, high volume*: Both databases share a box (and a single PostgreSQL instance) and source control has its own box.
- *DBA's choice, with Oracle*: Both databases share a box (and a single Oracle instance) and source control has its own box.

You can adopt one of these configurations, or you can make up a configuration of your own.

PostgreSQL or Oracle?

PostgreSQL 9.0 is installed automatically when you install TeamForge 6.1.1. Oracle 11 (R1 and R2) is also supported. If you intend to use Oracle, CollabNet recommends that you let the installer run its course, make sure things work normally, and then set up your Oracle database and switch over to it.

Choose your hardware

TeamForge can run on a wide range of hardware configurations.

- For a small team, you can install it on any laptop that can run VMware Player.
- In a large organization, you may need multi-processor hardware with NFS storage and multiple layers of redundancy.

Most sites will need something in between. For the minimal requirements, see *Hardware requirements for CollabNet TeamForge 6.1.1* on page 96.

You're on Red Hat Enterprise Linux 5.6, right?

Installing TeamForge on Red Hat is very similar to installing on any of the other supported environments, but there are differences that can throw you off. As you go through these steps, it's worth stopping occasionally to make sure the page you are reading corresponds to the platform you are using.

> 👉 **Note:** TeamForge 6.1.1 runs on SuSE Linux Enterprise Server 11 SP1 as well as Red Hat Enterprise Linux 5.6 and CentOS 5.6. Red Hat 5.4 and CentOS 5.4 are also supported, but we recommend using the latest supported version. You can use either the 64-bit or the 32-bit architecture of any supported OS.

Set up networking for your TeamForge box

After installing the operating system, prepare the networking connections and configuration that your TeamForge 6.1.1 site will require.

> 👉 **Note:** You must have root access to all the hosts you will be setting for your site.

1. Open the appropriate ports, and close all other ports.

   > 👉 **Note:** Expose only the JBOSS and Tomcat ports that are required for integration with another application, and open them only to that specific host IP address, even within your internal network.

   For detailed port requirements, see *What are the minimum ports to keep open for a TeamForge site?* on page 154

2. Use the `hostname` command to verify that the machine name is resolvable on the network.

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

3. 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.

4. If you are installing behind a proxy, specify your proxy settings.

```
export
http_proxy=http://<PROXY_USERNAME>:<PROXY_PASSWD>@<PROXY_HOST>:<PROXY_PORT>
```

5. If any mail service is running on port 25, stop it and make sure it won't restart.

   For example:

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

6. Use a tool such as Nessus to scan your server for potential vulnerabilities.

   (See *What are the minimum ports to keep open for a TeamForge site?* on page 154 for detailed security recommendations.)

Install TeamForge 6.1.1 the easy way

The easiest way to install TeamForge is to install on a single box, dedicated to TeamForge only, taking the default configuration settings. We call this a "dedicated" install.

☞ **Important:** It is *critical* that you start with a fresh machine, with no software installed apart from the boot loader. You must have root access to the server.

1. Install Red Hat Enterprise Linux 5.6 and log in as root.

   • The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 6.1.1* on page 96 for the full platform requirements.
   • See *the Red Hat installation guide* for help.

   ☞ **Important:** Don't customize your installation. Select only the default packages list.

2. Check your basic networking setup.

   See *Set up networking for your TeamForge box* on page 8 for details.

3. Download the TeamForge 6.1.1 installation repository from *open.collab.net*. Copy it to `/etc/yum.repos.d/`.

4. Install the TeamForge application.

```
yum install teamforge-app teamforge-scm teamforge-database teamforge-etl
```

   ☞ **Tip:** If the yum installer balks, you may have duplicate rpm packages.

   1. Get the `yum-utils` package, if it isn't already installed.

```
yum install yum-utils
```

   2. Check for duplicate packages.

```
package-cleanup --dupes
```

   3. Clean up the older packages, if any.

```
package-cleanup --cleandupes
```

4. Rerun the yum installer.

```
yum install teamforge [options]
```

5. Set up your site's master configuration file.

```
vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
```

☞ **Note:** `vi` is given as an example. Emacs, gedit, or any other *nix text editor will also work.

At this stage, we'll make the minimal modifications required to set up a usable TeamForge 6.1.1 site.

a) Edit the *DOMAIN_localhost* variable to point the site's URL to the server where the site is running.
   Use this format:

   ```
   DOMAIN_localhost=<domain_name>
   ```

   For example, suppose SuperVillain Inc. wants its development site to have the URL
   `https://worlddomination.supervillain.org`. The *DOMAIN_localhost* variable will look like this:

   ```
   DOMAIN_localhost=worlddomination.supervillain.org
   ```

b) Review the *DATABASE* variables.
   You may want to substitute your own values for the default database names, user names and passwords.

c) Turn on the new site-wide reporting functionality by adding the *REPORTS_** variables.
   See *Turn on site-wide reporting* on page 86 for details.

d) If you don't specify a domain name, replace the HOST_localhost token with the hostname. Otherwise, ViewVC
   pages for Subversion and CVS repositories created after the site is up, will be rendered with a CSS error.

e) ☞ **Important: It is mandatory that you include the *SCM_DEFAULT_SHARED_SECRET* token in
   the site-options.conf file of the primary TeamForge server, and give it a value of 16-24 characters.**

   Remember that you need to use that same key in the external SCM integration server also.

6. Run the installer.

   ```
   cd /opt/collabnet/teamforge-installer/6.1.1.0
   ./install.sh -r -I -V
   ```

7. Set up the initial site data.

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

8. Start TeamForge.

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

9. Apply some finishing touches and make sure everything is running smoothly.

a) Turn on SSL for your site by editing the relevant variables in the `site-options.conf` file.
   See *Set up SSL for your TeamForge site* on page 81.

b) Reboot the server and make sure all services come up automatically at startup.

c) Log into your site as the administrator.
   The value of the *DOMAIN* variable in the `site-options.conf` file is the URL to log into.

d) Install a project template.

TeamForge comes with a sample project template that showcases some of the platform's most interesting features. Site administrators and project managers can use this template to jump-start projects without a lot of manual setup steps. See *Install project templates* on page 78.

e) Create a sample project.

See *Create a TeamForge project*.

f) Write a welcome message to your site's users.

See *Create a site-wide broadcast*.

Install TeamForge the advanced way

If you need maximum flexibility and control over your TeamForge site, you'll want an "advanced" install.

Your TeamForge site consists of a collection of services that work together. You can host these services on one box or on different boxes, in whatever combination works best for your conditions.

In an "advanced" install, you'll identify the hosts on which the various components of your TeamForge site will run. For each machine that's part of your site, you'll set up the needed services and define how and where each service runs, and how they communicate with each other.

In principle, a multi-box TeamForge site can have its services running in a wide variety of combinations on an undefined number of boxes. However, real-world sites tend to follow one of the following patterns, depending on the specific needs of the community of site users.

Option 1: Install all TeamForge 6.1.1 services on a single box

In Option 1, we install the database, the reporting service, source control, and the datamart (reporting database) on the main application box.

This is not the same as a dedicated install, even though it uses a single box. See *Do I need an advanced TeamForge installation?* on page 152

☞ **Note:** In this example, we will use a separate port for the reporting database. This can help improve the perceived speed of a site when database utilization is high. If you want the reporting database to share port 5432 with the site database, omit the *REPORTS_DATABASE_PORT* variable. See *Option 2: Install TeamForge 6.1.1 with source control on a separate box* on page 15 for an example.

1. Install Red Hat Enterprise Linux 5.6 and log in as root.

   • The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 6.1.1* on page 96 for the full platform requirements.
   • See *the Red Hat installation guide* for help.

   ☞ **Important:** Don't customize your installation. Select only the default packages list.

2. Check your basic networking setup.

   See *Set up networking for your TeamForge box* on page 8 for details.

3. Download the TeamForge 6.1.1 installation repository from *open.collab.net*. Copy it to `/etc/yum.repos.d/`.

4. Install the TeamForge application.

   ```
   yum install teamforge-app teamforge-scm teamforge-database teamforge-etl
   ```

   ☞ **Tip:** If the yum installer balks, you may have duplicate rpm packages.

   1. Get the `yum-utils` package, if it isn't already installed.

      ```
      yum install yum-utils
      ```

   2. Check for duplicate packages.

```
package-cleanup --dupes
```

3. Clean up the older packages, if any.

```
package-cleanup --cleandupes
```

4. Rerun the yum installer.

```
yum install teamforge-app teamforge-scm teamforge-database
teamforge-etl
```

5. Rename the sample site configuration file from the installation package.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
cp conf/site-options-advanced.conf conf/site-options.conf
```

👉 **Note:** The files `site-options-small.conf`, `site-options-medium.conf` and `site-options-large.conf` contain options to tune the performance of the TeamForge site. To tune your site's performance, you can look through these files for the load specifications they are intended for, and use the appropriate one for your site's requirements.

6. Configure the site-options.conf file.

```
vi conf/site-options.conf
```

a) Identify the box and the services running on it.

```
HOST_localhost=app etl database datamart subversion cvs
DOMAIN_localhost=<myappbox.domain.com>
```

b) Configure the database and the datamart, substituting your own values for the defaults where appropriate.

👉 **Tip:** For more information about configuring variables, see *site-options.conf* on page 112.

```
DATABASE_TYPE=postgresql
DATABASE_USERNAME=ctfuser
DATABASE_PASSWORD=ctfpwd
DATABASE_NAME=ctfdb
DATABASE_MAX_POOL_SIZE=100
REPORTS_DATABASE_USERNAME=ctfrptuser
REPORTS_DATABASE_PASSWORD=ctfrptpwd
REPORTS_DATABASE_NAME=ctfrptdb
```

These options are optional:

```
REPORTS_DATABASE_READ_ONLY_USER=ctfrptreadonly
REPORTS_DATABASE_READ_ONLY_PASSWORD=rptropwd
ETL_SOAP_SHARED_SECRET=<arbitrary_string>
SOAP_ANONYMOUS_SHARED_SECRET=<arbitrary_string>
REPORTS_DATABASE_PORT=5632
```

👉 **Tip:**

- The database name and user name values are arbitrary alphanumeric strings.

c) If you don't specify a domain name, replace the HOST_localhost token with the hostname. Otherwise, ViewVC pages for Subversion and CVS repositories created after the site is up, will be rendered with a CSS error.

d) 👉 **Important: It is mandatory that you include the *SCM_DEFAULT_SHARED_SECRET* token in the site-options.conf file of the primary TeamForge server, and give it a value of 16-24 characters.**

Remember that you need to use that same key in the external SCM integration server also.

e) Don't forget to save the file.

7. Run the installer.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
./install.sh -r -I -V
```

8. Set up the site database.

a) Point the database to the local machine.

```
su - postgres
/usr/pgsql-9.0/bin/initdb -D /var/lib/pgsql/9.0/data
vi /var/lib/pgsql/9.0/data/postgresql.conf
```

```
listen_addresses = '127.0.0.1,<IP address of database box>'
```

b) Configure database access for the site database.

```
vi /var/lib/pgsql/9.0/data/pg_hba.conf
```

```
local   all                     all
            trust
# IPv4 local connections:
host    all                     all                         127.0.0.1/32
            trust
# IPv6 local connections:
#host   all                     all                         ::1/128
            trust
host    <DATABASE_NAME>  <DATABASE_USERNAME       <IP address of app box>/32
    md5
```

9. Set up the reporting database (datamart).

> ☞ **Note:** You can skip this step if one of these is true:
>
> - You are not providing site-wide reporting.
> - You are providing reporting, and your reporting database will share port 5432 with your site database. In this case, just add these two lines to /var/lib/pgsql/9.0/data/pg_hba.conf:
>
> ```
> host <REPORTS_DATABASE_NAME> <REPORTS_DATABASE_USERNAME>                <IP
>  address of this box>/32   md5
> host <REPORTS_DATABASE_NAME> <REPORTS_DATABASE_READ_ONLY_USERNAME> <IP
>  address of this box>/32   md5
> ```

a) Point the datamart to the local machine.

```
/usr/pgsql-9.0/bin/initdb -D /var/lib/pgsql/9.0/reports
vi /var/lib/pgsql/9.0/reports/postgresql.conf
```

```
listen_addresses = '127.0.0.1,<IP address of datamart box>'
port=5632
```

b) Configure access for the datamart.

```
vi /var/lib/pgsql/9.0/reports/pg_hba.conf
```

```
local all                      all
```

```
                              trust
# IPv4 local connections:
host  all                          all
127.0.0.1/32                     trust
# IPv6 local connections:
#host all                          all                              ::1/128
                     trust
host  <REPORTS_DATABASE_NAME> <REPORTS_DATABASE_USERNAME>          <IP
address of this box>/32  md5
host  <REPORTS_DATABASE_NAME> <REPORTS_DATABASE_READ_ONLY_USERNAME> <IP
address of this box>/32  md5
```

10. Start PostgreSQL.

```
exit
/etc/init.d/postgresql-9.0 start
```

11. Create the databases.

```
su - postgres
```

a) Site database:

```
createuser -P -S --createdb --no-createrole <ctfdatabase_username>
createdb -E UTF8 -O <ctfdatabase_username> <ctfdatabasename>
```

b) Reporting database:

```
createuser -p <reports_database_port> -P -S --createdb --no-createrole
<reports_database_username>
createuser -p <reports_database_port> -P -S --createdb --no-createrole
<reports_database_readonlyusername>
createdb -p <reports_database_port> -E UTF8 -O <reports database username>
<reports_database_name>
```

👉 **Note:**

- For the passwords, use the same passwords you recorded in the `site-options.conf` file.
- To specify a non-default port, add the -p option.

c) Restart PostgreSQL.

```
exit
/etc/init.d/postgresql-9.0 restart
```

12. Set up the initial site data.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
./bootstrap-data.sh
```

13. Swap in the new Apache configuration file.

```
cd /etc/httpd/conf
mv httpd.conf httpd.conf_old
cp httpd.conf.cn_new httpd.conf
/etc/init.d/httpd start
```

14. Start TeamForge.

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

☞ **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 600 seconds, giving up now. Please check the log:
/opt/collabnet/teamforge/log/apps/service.log FAILED
```

This can safely be ignored.

Your installation is complete. Go to *Verify your TeamForge 6.1.1 installation* on page 40 for recommended steps to get started with your site.

Option 2: Install TeamForge 6.1.1 with source control on a separate box

In Option 2, we install the site database, the datamart and the reporting service on the main application box and source control services on a separate box. This can help on a site with heavy source code commit or checkout traffic.

☞ **Note:** In this example, we will use the same port for the site database and the reports database. This is the default setting. When heavy traffic is expected, it can be a good idea to use port 5632 for the reporting database. See *Option 1* for an example.

Do this on the main TeamForge application server. We'll call this `my.app.box`.

1. Install Red Hat Enterprise Linux 5.6 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 6.1.1* on page 96 for the full platform requirements.
- See *the Red Hat installation guide* for help.

☞ **Important:** Don't customize your installation. Select only the default packages list.

2. Check your basic networking setup.

See *Set up networking for your TeamForge box* on page 8 for details.

3. Download the TeamForge 6.1.1 installation repository from *open.collab.net*. Copy it to `/etc/yum.repos.d/`.

4. Install the TeamForge application and reporting service.

```
yum install teamforge-app teamforge-database teamforge-etl
```

☞ **Tip:** If the yum installer balks, you may have duplicate rpm packages.

1. Get the `yum-utils` package, if it isn't already installed.

```
yum install yum-utils
```

2. Check for duplicate packages.

```
package-cleanup --dupes
```

3. Clean up the older packages, if any.

```
package-cleanup --cleandupes
```

4. Rerun the yum installer.

```
yum install teamforge-app teamforge-database
teamforge-etl
```

5. Rename the sample site configuration file from the installation package.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
cp conf/site-options-advanced.conf conf/site-options.conf
```

☞ **Note:** The files `site-options-small.conf`, `site-options-medium.conf` and `site-options-large.conf` contain options to tune the performance of the TeamForge site. To tune your site's performance, you can look through these files for the load specifications they are intended for, and use the appropriate one for your site's requirements.

6. Configure the site-options.conf file.

```
vi conf/site-options.conf
```

a) Identify the boxes and the services running on them.

```
HOST_localhost=app etl database datamart
DOMAIN_localhost=<myappbox.domain.com>
HOST_<mycodebox.domain.com>=subversion cvs
```

b) Configure the database and the datamart.

☞ **Tip:** For more information about configuring variables, see *site-options.conf* on page 112.

```
DATABASE_TYPE=postgresql
DATABASE_USERNAME=ctfuser
DATABASE_PASSWORD=ctfpwd
DATABASE_NAME=ctfdb
DATABASE_MAX_POOL_SIZE=100
REPORTS_DATABASE_USERNAME=ctfrptuser
REPORTS_DATABASE_PASSWORD=ctfrptpwd
REPORTS_DATABASE_NAME=ctfrptdb
REPORTS_DATABASE_READ_ONLY_USER=ctfrptreadonly
REPORTS_DATABASE_READ_ONLY_PASSWORD=rptropwd
ETL_SOAP_SHARED_SECRET=<arbitrary_string>
SOAP_ANONYMOUS_SHARED_SECRET=<arbitrary_string>
```

☞ **Tip:** All the database names, user names and passwords values are arbitrary alphanumeric strings.

c) If you don't specify a domain name, replace the HOST_localhost token with the hostname. Otherwise, ViewVC pages for Subversion and CVS repositories created after the site is up, will be rendered with a CSS error.

d) ☞ **Important: It is mandatory that you include the *SCM_DEFAULT_SHARED_SECRET* token in the site-options.conf file of the primary TeamForge server, and give it a value of 16-24 characters.**

Remember that you need to use that same key in the external SCM integration server also.

e) Don't forget to save the file.

7. Run the installer.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
./install.sh -r -I -V
```

8. Set up the site database.

a) Point the database to the local machine.

```
su - postgres
/usr/pgsql-9.0/bin/initdb -D /var/lib/pgsql/9.0/data/
```

```
vi /var/lib/pgsql/9.0/data/postgresql.conf
```

```
listen_addresses = '127.0.0.1,<IP address of database box>'
```

b) Configure database access.

```
vi /var/lib/pgsql/9.0/data/pg_hba.conf
```

```
    # "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>
  <IP address of my.app.box>/32     md5
    host    <REPORTS_DATABASE_NAME>     <REPORTS_DATABASE_USERNAME>
  <IP address of my.app.box>/32     md5
    host    <REPORTS_DATABASE_NAME>
<REPORTS_DATABASE_READ_ONLY_USERNAME> <IP address of my.app.box>/32    md5
```

```
exit
/etc/init.d/postgresql-9.0 start
```

9. Create the databases.

```
su - postgres
```

a) Site database:

```
createuser -P -S --createdb --no-createrole <ctfdatabase_username>
createdb -E UTF8 -O <ctfdatabase_username> <ctfdatabasename>
```

b) Reporting database:

```
createuser -p <reports_database_port> -P -S --createdb --no-createrole
<reports_database_username>
createuser -p <reports_database_port> -P -S --createdb --no-createrole
<reports_database_readonlyusername>
createdb -p <reports_database_port> -E UTF8 -O <reports database username>
<reports_database_name>
```

☞ **Note:**

- For the passwords, use the same passwords you recorded in the `site-options.conf` file.
- To specify a non-default port, add the -p option.

c) Restart PostgreSQL.

```
exit
/etc/init.d/postgresql-9.0 restart
```

10. Set up the initial site data.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
./bootstrap-data.sh
```

11. If you are installing on a machine that is behind a proxy, unset the *http_proxy* variable.

```
export http_proxy=
```

12. Swap in the new Apache configuration file.

```
cd /etc/httpd/conf
mv httpd.conf httpd.conf_old
cp httpd.conf.cn_new httpd.conf
/etc/init.d/httpd start
```

Do this on the source code server. We'll call this `my.code.box.`

13. Install Red Hat Enterprise Linux 5.6 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 6.1.1* on page 96 for the full platform requirements.
- See *the Red Hat installation guide* for help.

☞ **Important:** Don't customize your installation. Select only the default packages list.

14. Check your basic networking setup.

See *Set up networking for your TeamForge box* on page 8 for details.

15. Download the TeamForge 6.1.1 installation repository from *open.collab.net*. Copy it to /etc/yum.repos.d/.

16. Install the TeamForge source control service.

```
yum install teamforge-scm
```

☞ **Tip:** If the yum installer balks, you may have duplicate rpm packages.

1. Get the `yum-utils` package, if it isn't already installed.

```
yum install yum-utils
```

2. Check for duplicate packages.

```
package-cleanup --dupes
```

3. Clean up the older packages, if any.

```
package-cleanup --cleandupes
```

4. Rerun the yum installer.

```
yum install teamforge-scm
```

17. Rename the sample site configuration file from the installation package.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
cp conf/site-options-advanced.conf conf/site-options.conf
```

☞ **Note:** The files `site-options-small.conf`, `site-options-medium.conf` and `site-options-large.conf` contain options to tune the performance of the TeamForge site. To tune your site's performance, you can look through these files for the load specifications they are intended for, and use the appropriate one for your site's requirements.

18. Configure the site-options.conf file.

a) Identify the boxes and the services running on them.

```
NODE_NAME=localhost
HOST_localhost=subversion cvs
DOMAIN_localhost=<mycodebox.domain.com>
HOST_<myappbox.domain.com>=app database
```

b) Configure the database connection.

```
DATABASE_TYPE=postgresql
DATABASE_USERNAME=ctfuser
DATABASE_PASSWORD=ctfpwd
DATABASE_NAME=ctfdb
```

c) If you don't specify a domain name, replace the HOST_localhost token with the hostname. Otherwise, ViewVC pages for Subversion and CVS repositories created after the site is up, will be rendered with a CSS error.

d) Use the shared secret key, *SCM_DEFAULT_SHARED_SECRET*, from the `site-options.conf` file on the primary TeamForge server.

e) Don't forget to save the file.

19. Re-create the runtime environment.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
./install.sh -r -I -V -b
```

20. If you are installing on a machine that is behind a proxy, unset the *http_proxy* variable.

```
export http_proxy=
```

21. Swap in the new Apache configuration file.

```
cd /etc/httpd/conf
mv httpd.conf httpd.conf_old
cp httpd.conf.cn_new httpd.conf
/etc/init.d/httpd start
```

22. Start TeamForge.

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

Do this on `my.app.box`:

23. Start TeamForge.

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

☞ **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 600 seconds, giving up now. Please check the log:
/opt/collabnet/teamforge/log/apps/service.log FAILED
```

This can safely be ignored.

Your installation is complete. Go to *Verify your TeamForge 6.1.1 installation* on page 40 for recommended steps to get started with your site.

Option 3: Install TeamForge 6.1.1 with reporting data on a separate box
In Option 3, we install the database, the reporting service and source control on the main application box and the datamart (reporting database) on a separate box. This enables site administrators to monitor activity closely without overloading the rest of the site.

In this example, we will specify a separate port for the reports database. By default, both the site database and the reporting database use port 5432, but when heavy traffic is expected, it can be a good idea to use port 5632 for the reporting database.

☞ **Note:** If the remote machine where the datamart will run is not under your direct control, check with the DBA or other person in charge of that resource to make sure you can carry out these instructions on that box.

Do this on the main TeamForge application server. We'll call this `my.app.box`.

1. Install Red Hat Enterprise Linux 5.6 and log in as root.

    • The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 6.1.1* on page 96 for the full platform requirements.
    • See *the Red Hat installation guide* for help.

    ☞ **Important:** Don't customize your installation. Select only the default packages list.

2. Check your basic networking setup.
    See *Set up networking for your TeamForge box* on page 8 for details.

3. Download the TeamForge 6.1.1 installation repository from *open.collab.net*. Copy it to `/etc/yum.repos.d/`.

4. Install the TeamForge application and the database, reporting and source control services.

    ```
    yum install teamforge-app teamforge-database teamforge-etl teamforge-scm
    ```

    ☞ **Tip:** If the yum installer balks, you may have duplicate rpm packages.

    1. Get the `yum-utils` package, if it isn't already installed.

        ```
        yum install yum-utils
        ```

    2. Check for duplicate packages.

        ```
        package-cleanup --dupes
        ```

    3. Clean up the older packages, if any.

        ```
        package-cleanup --cleandupes
        ```

    4. Rerun the yum installer.

        ```
        yum install teamforge-app teamforge-database teamforge-etl
        teamforge-scm
        ```

5. Rename the sample site configuration file from the installation package.

    ```
    cd /opt/collabnet/teamforge-installer/6.1.1.0
    cp conf/site-options-advanced.conf conf/site-options.conf
    ```

    ☞ **Note:** The files `site-options-small.conf`, `site-options-medium.conf` and `site-options-large.conf` contain options to tune the performance of the TeamForge site. To tune your site's performance, you can look through these files for the load specifications they are intended for, and use the appropriate one for your site's requirements.

6. Configure the `site-options.conf` file.

```
vi conf/site-options.conf
```

a) Identify the boxes and the services running on them.

```
HOST_localhost=app database subversion cvs etl
DOMAIN_localhost=<myappbox.domain.com>
HOST_<mydatamartbox.domain.com>=datamart
```

b) Configure the database and the datamart.

☞ **Tip:** For more information about configuring variables, see *site-options.conf* on page 112.

```
DATABASE_TYPE=postgresql
DATABASE_USERNAME=ctfuser
DATABASE_PASSWORD=ctfpwd
DATABASE_NAME=ctfdb
DATABASE_MAX_POOL_SIZE=100
REPORTS_DATABASE_USERNAME=ctfrptuser
REPORTS_DATABASE_PASSWORD=ctfrptpwd
REPORTS_DATABASE_NAME=ctfrptdb
REPORTS_DATABASE_READ_ONLY_USER=ctfrptreadonly
REPORTS_DATABASE_READ_ONLY_PASSWORD=rptropwd
ETL_SOAP_SHARED_SECRET=<arbitrary_string>
SOAP_ANONYMOUS_SHARED_SECRET=<arbitrary_string>
REPORTS_DATABASE_PORT=5632
```

☞ **Tip:** The database name and user name values are arbitrary alphanumeric strings.

c) If you don't specify a domain name, replace the HOST_localhost token with the hostname. Otherwise, ViewVC pages for Subversion and CVS repositories created after the site is up, will be rendered with a CSS error.

d) ☞ **Important: It is mandatory that you include the *SCM_DEFAULT_SHARED_SECRET* token in the site-options.conf file of the primary TeamForge server, and give it a value of 16-24 characters.**

Remember that you need to use that same key in the external SCM integration server also.

e) Don't forget to save the file.

7. Run the installer.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
./install.sh -r -I -V
```

8. Set up the site database.

a) Point the database to the local machine.

```
su - postgres
/usr/pgsql-9.0/bin/initdb -D /var/lib/pgsql/9.0/data
vi /var/lib/pgsql/9.0/data/postgresql.conf
listen_addresses = '127.0.0.1,<IP address of database box>'
```

b) Configure database access.

```
vi /var/lib/pgsql/9.0/data/pg_hba.conf
```

```
local   all                     all
    trust
```

```
# IPv4 local connections:
host    all                       all                         127.0.0.1/32
    trust
# IPv6 local connections:
#host   all                       all                         ::1/128
    trust
host    <DATABASE_NAME>     <DATABASE_USERNAME>         <IP address of
my.app.box>/32    md5
```

```
exit
/etc/init.d/postgresql-9.0 start
```

c) Create the site database.

```
su - postgres
createuser -P -S --createdb --no-createrole <ctfdatabase_username>
createdb -E UTF8 -O <ctfdatabase_username> <ctfdatabasename>
exit
/etc/init.d/postgresql-9.0 restart
```

9. If you are installing on a machine that is behind a proxy, unset the *http_proxy* variable.

```
export http_proxy=
```

Do this on the datamart server. We'll call this `my.datamart.box`.

10. Install Red Hat Enterprise Linux 5.6 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 6.1.1* on page 96 for the full platform requirements.
- See *the Red Hat installation guide* for help.

☞ **Important:** Don't customize your installation. Select only the default packages list.

11. Install PostgreSQL 9.0.

See *the PostgreSQL docs* for support.

12. Set up the reporting database (datamart).

a) Point the database to the local machine.

```
su - postgres
/usr/pgsql-9.0/bin/initdb -D /var/lib/pgsql/9.0/reports
vi /var/lib/pgsql/9.0/reports/postgresql.conf

    listen_addresses = '127.0.0.1,<IP address of my.datamart.box>'
    port=5632
```

b) Configure access for the datamart.

```
vi /var/lib/pgsql/9.0/reports/pg_hba.conf
```

```
local   all                       all
                        trust
# IPv4 local connections:
host    all                       all
   127.0.0.1/32           trust
# IPv6 local connections:
#host   all                       all
   ::1/128                trust
```

```
host    <REPORTS_DATABASE_NAME>         <REPORTS_DATABASE_USERNAME>
  <IP address of my.app.box>/32    md5
host    <REPORTS_DATABASE_NAME>         <REPORTS_DATABASE_READ_ONLY_USER>
  <IP address of my.app.box>/32    md5
```

```
   exit
```

c) Copy the datamart control script from `/opt/collabnet/teamforge/runtime/scripts` on `my.app.box` to the `/tmp` directory of `my.datamart.box`.

```
mv /etc/init.d/postgresql-9.0 /etc/init.d/postgresql-9.0_orig
cp /tmp/postgresql_reports /etc/init.d
```

👉 **Note:** The `postgresql_reports` script is the one to use only if the datamart is running on a different server from the database, as it is in this example. In all other cases, use the `postgresql` script.

d) Start the reporting database service.

```
/etc/init.d/postgresql-9.0_reports start
```

e) Log in as the `postgres` user and create the reporting database.
Use the values from your `site-options.conf` file.

```
su - postgres
createuser -p <reports_database_port> -P -S --createdb --no-createrole
<reports_database_username>
createuser -p <reports_database_port> -P -S --createdb --no-createrole
<reports_database_readonlyusername>
createdb -p <reports_database_port> -E UTF8 -O <reports database username>
<reports database name>
exit
/etc/init.d/postgresql-9.0_reports restart
```

👉 **Note:** To specify a non-default port, add the -p option.

Do this on `my.app.box`:

13. Set up the initial site data.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
./bootstrap-data.sh
```

14. Swap in the new Apache configuration file.

```
cd /etc/httpd/conf
mv httpd.conf httpd.conf_old
cp httpd.conf.cn_new httpd.conf
/etc/init.d/httpd start
```

15. Start TeamForge.

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

👉 **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 600 seconds, giving up now. Please check the log:
/opt/collabnet/teamforge/log/apps/service.log FAILED
```

This can safely be ignored.

Your installation is complete. Go to *Verify your TeamForge 6.1.1 installation* on page 40 for recommended steps to get started with your site.

Option 4: Distribute TeamForge 6.1.1 on three boxes
In Option 4, we install the site database and the reporting service on the main application box, while source control and the reporting database each get their own box. This can be a way to support the requirements of both coders and management.

In this example, we will use the same port for the site database and the reports database. This is the default setting. When heavy traffic is expected, it can be a good idea to use port 5632 for the reporting database. See *Option 1* for an example.

☞ **Note:** If either of the remote machines (the datamart or the source code box) is not under your direct control, check with the DBA or other person in charge of that resource to make sure you can carry out these instructions on that box.

Do this on the main TeamForge application server. We'll call this `my.app.box`.

1. Install Red Hat Enterprise Linux 5.6 and log in as root.

    • The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 6.1.1* on page 96 for the full platform requirements.
    • See *the Red Hat installation guide* for help.

    ☞ **Important:** Don't customize your installation. Select only the default packages list.

2. Check your basic networking setup.

    See *Set up networking for your TeamForge box* on page 8 for details.

3. Download the TeamForge 6.1.1 installation repository from *open.collab.net*. Copy it to /etc/yum.repos.d/.

4. Install the TeamForge application database and reporting service.

    ```
    yum install teamforge-app teamforge-database teamforge-etl
    ```

    ☞ **Tip:** If the yum installer balks, you may have duplicate rpm packages.

    1. Get the `yum-utils` package, if it isn't already installed.

        ```
        yum install yum-utils
        ```

    2. Check for duplicate packages.

        ```
        package-cleanup --dupes
        ```

    3. Clean up the older packages, if any.

        ```
        package-cleanup --cleandupes
        ```

    4. Rerun the yum installer.

        ```
        yum install teamforge-app teamforge-database
        teamforge-etl
        ```

5. Rename the sample site configuration file from the installation package.

    ```
    cd /opt/collabnet/teamforge-installer/6.1.1.0
    cp conf/site-options-advanced.conf conf/site-options.conf
    ```

👉 **Note:** The files `site-options-small.conf`, `site-options-medium.conf` and `site-options-large.conf` contain options to tune the performance of the TeamForge site. To tune your site's performance, you can look through these files for the load specifications they are intended for, and use the appropriate one for your site's requirements.

6. Configure the site-options.conf file.

```
vi conf/site-options.conf
```

a) Identify your boxes and the services running on them.

```
HOST_localhost=app etl database
DOMAIN_localhost=<myappbox.domain.com>
HOST_<mydatamartbox.domain.com>=datamart
HOST_<mycodebox.domain.com>=subversion cvs
```

b) Configure the database and the datamart.

👉 **Tip:** For more information about configuring variables, see *site-options.conf* on page 112.

```
DATABASE_TYPE=postgresql
DATABASE_USERNAME=ctfuser
DATABASE_PASSWORD=ctfpwd
DATABASE_NAME=ctfdb
DATABASE_MAX_POOL_SIZE=100
REPORTS_DATABASE_USERNAME=ctfrptuser
REPORTS_DATABASE_PASSWORD=ctfrptpwd
REPORTS_DATABASE_NAME=ctfrptdb
REPORTS_DATABASE_READ_ONLY_USER=ctfrptreadonly
REPORTS_DATABASE_READ_ONLY_PASSWORD=rptropwd
ETL_SOAP_SHARED_SECRET=<arbitrary_string>
SOAP_ANONYMOUS_SHARED_SECRET=<arbitrary_string>
```

👉 **Tip:** The database name and user name values are arbitrary alphanumeric strings.

c) If you don't specify a domain name, replace the HOST_localhost token with the hostname. Otherwise, ViewVC pages for Subversion and CVS repositories created after the site is up, will be rendered with a CSS error.

d) 👉 **Important: It is mandatory that you include the *SCM_DEFAULT_SHARED_SECRET* token in the site-options.conf file of the primary TeamForge server, and give it a value of 16-24 characters.**

Remember that you need to use that same key in the external SCM integration server also.

e) Don't forget to save the file.

7. Run the installer.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
./install.sh -r -I -V
```

8. Set up the site database.

a) Point the database to the local machine.

```
su - postgres
/usr/pgsql-9.0/bin/initdb -D /var/lib/pgsql/9.0/data/
vi /var/lib/pgsql/9.0/data/postgresql.conf
```

```
listen_addresses = '127.0.0.1,<IP address of database box>'
```

b) Configure database access.

```
vi /var/lib/pgsql/9.0/data/pg_hba.conf
```

```
    local   all          all                              trust
    # IPv4 local connections:
    host    all          all         127.0.0.1/32         trust
    # IPv6 local connections:
    #host   all          all          ::1/128             trust
    host    <DATABASE_NAME>            <DATABASE_USERNAME>
  <IP address of my.app.box>/32    md5
```

```
exit
/etc/init.d/postgresql-9.0 start
```

c) Log in as the `postgres` user and create the site database.

```
su - postgres
createuser -P -S --createdb --no-createrole <ctfdatabase_username>
createdb -E UTF8 -O <ctfdatabase_username> <ctfdatabasename>
exit
```

9. If you are installing on a machine that is behind a proxy, unset the *http_proxy* variable.

```
export http_proxy=
```

Do this on the datamart server. We'll call this `my.datamart.box`.

10. Install Red Hat Enterprise Linux 5.6 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 6.1.1* on page 96 for the full platform requirements.
- See *the Red Hat installation guide* for help.

☞   **Important:** Don't customize your installation. Select only the default packages list.

11. Install PostgreSQL 9.0.

See *the PostgreSQL docs* for support.

12. Set up the reporting database.

a) Point the database to the local machine.

```
su - postgres
/usr/pgsql-9.0/bin/initdb -D /var/lib/pgsql/9.0/reports
vi /var/lib/pgsql/9.0/reports/postgresql.conf
```

```
listen_addresses = '127.0.0.1,<IP address of my.datamart.box>'
```

b) Configure database access.

```
vi /var/lib/pgsql/9.0/reports/pg_hba.conf
```

```
    local   all          all                              trust
    # IPv4 local connections:
    host    all          all         127.0.0.1/32         trust
    # IPv6 local connections:
    #host   all          all          ::1/128             trust
    host    <REPORTS_DATABASE_NAME>    <REPORTS_DATABASE_USERNAME>
  <IP address of my.app.box>/32    md5
    host    <REPORTS_DATABASE_NAME>
<REPORTS_DATABASE_READ_ONLY_USERNAME> <IP address of my.app.box>/32    md5
```

```
exit
```

c) Copy the datamart control script from `/opt/collabnet/teamforge/runtime/scripts` on `my.app.box` to the `/tmp` directory of `my.datamart.box`.

```
mv /etc/init.d/postgresql-9.0 /etc/init.d/postgresql-9.0_orig
cp /tmp/postgresql_reports /etc/init.d
```

☞ **Note:** The `postgresql_reports` script is the one to use only if the datamart is running on a different server from the database, as it is in this example. In all other cases, use the `postgresql` script.

d) Start the reporting database service.

```
/etc/init.d/postgresql-9.0_reports start
```

e) Log in as the `postgres` user and create the reporting database.
Use the values from your `site-options.conf` file.

```
su - postgres
createuser -p <reports_database_port> -P -S --createdb --no-createrole
<reports_database_username>
createuser -p <reports_database_port> -P -S --createdb --no-createrole
<reports_database_readonlyusername>
createdb -p <reports_database_port> -E UTF8 -O <reports database username>
<reports database name>
exit
/etc/init.d/postgresql-9.0_reports restart
```

☞ **Note:** To specify a non-default port, add the -p option.

Do this on `my.app.box`:

13. Set up the initial site data.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
./bootstrap-data.sh
```

14. Swap in the new Apache configuration file.

```
cd /etc/httpd/conf
mv httpd.conf httpd.conf_old
cp httpd.conf.cn_new httpd.conf
/etc/init.d/httpd start
```

Do this on the source control server. We'll call this `my.code.box`.

15. Install Red Hat Enterprise Linux 5.6 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 6.1.1* on page 96 for the full platform requirements.
- See *the Red Hat installation guide* for help.

☞ **Important:** Don't customize your installation. Select only the default packages list.

16. Check your basic networking setup.
See *Set up networking for your TeamForge box* on page 8 for details.

17. Download the TeamForge 6.1.1 installation repository from *open.collab.net*. Copy it to `/etc/yum.repos.d/`.

18. Install the TeamForge application.

```
yum install teamforge-scm
```

☞ **Tip:** If the yum installer balks, you may have duplicate rpm packages.

1. Get the `yum-utils` package, if it isn't already installed.

```
yum install yum-utils
```

2. Check for duplicate packages.

```
package-cleanup --dupes
```

3. Clean up the older packages, if any.

```
package-cleanup --cleandupes
```

4. Rerun the yum installer.

```
yum install teamforge-scm
```

19. Rename the sample site configuration file from the installation package.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
cp conf/site-options-advanced.conf conf/site-options.conf
```

☞ **Note:** The files `site-options-small.conf`, `site-options-medium.conf` and `site-options-large.conf` contain options to tune the performance of the TeamForge site. To tune your site's performance, you can look through these files for the load specifications they are intended for, and use the appropriate one for your site's requirements.

20. Configure the site-options.conf file.

a) Identify the boxes and the services running on them.

```
NODE_NAME=localhost
HOST_localhost=subversion cvs
DOMAIN_localhost=<mycodebox.domain.com>
HOST<myappbox.domain.com>=app database
```

b) Configure the database connection.

```
DATABASE_TYPE=postgresql
DATABASE_USERNAME=ctfuser
DATABASE_PASSWORD=ctfpwd
DATABASE_NAME=ctfdb
```

c) If you don't specify a domain name, replace the HOST_localhost token with the hostname. Otherwise, ViewVC pages for Subversion and CVS repositories created after the site is up, will be rendered with a CSS error.

d) Use the shared secret key, *SCM_DEFAULT_SHARED_SECRET*, from the `site-options.conf` file on the primary TeamForge server.

21. Run the installer.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
./install.sh -r -I -V -b
```

22. If you are installing on a machine that is behind a proxy, unset the *http_proxy* variable.

```
export http_proxy=
```

23. Swap in the new Apache configuration file.

```
cd /etc/httpd/conf
mv httpd.conf httpd.conf_old
cp httpd.conf.cn_new httpd.conf
/etc/init.d/httpd start
```

24. Start TeamForge.

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

Do this on `my.app.box`:

25. Start TeamForge.

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

☞ **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 600 seconds, giving up now. Please check the log:
/opt/collabnet/teamforge/log/apps/service.log FAILED
```

This can safely be ignored.

Your installation is complete. Go to *Verify your TeamForge 6.1.1 installation* on page 40 for recommended steps to get started with your site.

Option 5: Distribute TeamForge 6.1.1 services a different way

In Option 5, we install the reporting service on the main application box, both databases on another box, and source control on a third box. This may be appropriate when intensive database utilization is a factor.

In this example, we will specify a separate port for the reports database. By default, both the site database and the reporting database use port 5432, but when heavy traffic is expected, it can be a good idea to use port 5632 for the reporting database.

☞ **Note:** If either of the remote machines (the data box or the source code box) is not under your direct control, check with the DBA or other person in charge of that resource to make sure you can carry out these instructions on that box.

Do this on the main TeamForge application server. We'll call this `my.app.box`.

1. Install Red Hat Enterprise Linux 5.6 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 6.1.1* on page 96 for the full platform requirements.
- See *the Red Hat installation guide* for help.

☞ **Important:** Don't customize your installation. Select only the default packages list.

2. Check your basic networking setup.

See *Set up networking for your TeamForge box* on page 8 for details.

3. Download the TeamForge 6.1.1 installation repository from *open.collab.net*. Copy it to /etc/yum.repos.d/.

4. Install the TeamForge application and reporting service.

```
yum install teamforge-app teamforge-etl
```

☞ **Tip:** If the yum installer balks, you may have duplicate rpm packages.

   1. Get the `yum-utils` package, if it isn't already installed.

```
yum install yum-utils
```

2. Check for duplicate packages.

```
package-cleanup --dupes
```

3. Clean up the older packages, if any.

```
package-cleanup --cleandupes
```

4. Rerun the yum installer.

```
yum install teamforge-app teamforge-etl
```

5. Rename the sample site configuration file from the installation package.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
cp conf/site-options-advanced.conf conf/site-options.conf
```

☞ **Note:** The files `site-options-small.conf`, `site-options-medium.conf` and `site-options-large.conf` contain options to tune the performance of the TeamForge site. To tune your site's performance, you can look through these files for the load specifications they are intended for, and use the appropriate one for your site's requirements.

6. Configure the `site-options.conf` file.

```
vi conf/site-options.conf
```

a) Identify the boxes and the services running on them.

```
HOST_localhost=app etl
DOMAIN_localhost=<myappbox.domain.com>
HOST_<mydbbox.domain.com>=database datamart
HOST_<myscmbox.domain.com>=subversion cvs
```

b) Configure the database and the datamart.

☞ **Tip:** For more information about configuring variables, see *site-options.conf* on page 112.

```
DATABASE_TYPE=postgresql
DATABASE_USERNAME=ctfuser
DATABASE_PASSWORD=ctfpwd
DATABASE_NAME=ctfdb
DATABASE_MAX_POOL_SIZE=100
REPORTS_DATABASE_USERNAME=ctfrptuser
REPORTS_DATABASE_PASSWORD=ctfrptpwd
REPORTS_DATABASE_NAME=ctfrptdb
REPORTS_DATABASE_READ_ONLY_USER=ctfrptreadonly
REPORTS_DATABASE_READ_ONLY_PASSWORD=rptropwd
ETL_SOAP_SHARED_SECRET=<arbitrary_string>
SOAP_ANONYMOUS_SHARED_SECRET=<arbitrary_string>
REPORTS_DATABASE_PORT=5632
```

☞ **Tip:** The database name and user name values are arbitrary alphanumeric strings.

c) If you don't specify a domain name, replace the HOST_localhost token with the hostname. Otherwise, ViewVC pages for Subversion and CVS repositories created after the site is up, will be rendered with a CSS error.

d) 👉   **Important:** **It is mandatory that you include the *SCM_DEFAULT_SHARED_SECRET* token in the site-options.conf file of the primary TeamForge server, and give it a value of 16-24 characters.**

Remember that you need to use that same key in the external SCM integration server also.

e) Don't forget to save the file.

7. If you are installing on a machine that is behind a proxy, unset the *http_proxy* variable.

```
export http_proxy=
```

8. Run the installer.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
./install.sh -r -I -V
```

Do this on the database/datamart server. We'll call this `my.data.box`.

9. Install Red Hat Enterprise Linux 5.6 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 6.1.1* on page 96 for the full platform requirements.
- See *the Red Hat installation guide* for help.

👉   **Important:** Don't customize your installation. Select only the default packages list.

10. Install PostgreSQL 9.0.

See *the PostgreSQL docs* for support.

11. Set up the site database.

a) Point the database to the local machine.

```
su - postgres
/usr/pgsql-9.0/bin/initdb -D /var/lib/pgsql/9.0/data
vi /var/lib/pgsql/9.0/data/postgresql.conf
```

```
listen_addresses = '127.0.0.1,<IP address of database box>'
```

b) Configure database access.

```
vi /var/lib/pgsql/9.0/data/pg_hba.conf
```

```
local   all                     all
    trust
# IPv4 local connections:
host    all                     all                     127.0.0.1/32
    trust
# IPv6 local connections:
#host   all                     all                     ::1/128
    trust
host    <DATABASE_NAME>  <DATABASE_USERNAME>     <IP address of
my.app.box>/32   md5
```

12. Set up the reporting database (datamart).

a) Point the database to the local machine.

```
/usr/pgsql-9.0/bin/initdb -D /var/lib/pgsql/9.0/reports
```

```
vi /var/lib/pgsql/9.0/reports/postgresql.conf
```

```
listen_addresses = '127.0.0.1,<IP address of my datamart box>'
      port=5632
```

b) Configure access for the datamart.

```
vi /var/lib/pgsql/9.0/reports/pg_hba.conf
```

```
local    all                              all
                                                 trust
# IPv4 local connections:
host     all                              all
   127.0.0.1/32                                  trust
# IPv6 local connections:
#host    all                              all
   ::1/128                                       trust
host    <REPORTS_DATABASE_NAME>    <REPORTS_DATABASE_USERNAME>          <IP
 address of my.app.box/32    md5
host    <REPORTS_DATABASE_NAME>    <REPORTS_DATABASE_READ_ONLY_USERNAME>
<IP address of my.app.box>/32    md5
```

```
exit
```

c) Copy the postgresql script from `/opt/collabnet/teamforge/runtime/scripts` on `my.app.box` to the `/tmp` directory of `my.data.box`.

```
mv /etc/init.d/postgresql-9.0 /etc/init.d/postgresql-9.0_orig
cp /tmp/postgresql /etc/init.d
```

d) Start the reporting database service.

```
/etc/init.d/postgresql-9.0 start
```

13. Create the databases.

```
su - postgres
```

a) Site database:

```
createuser -P -S --createdb --no-createrole <ctfdatabase_username>
createdb -E UTF8 -O <ctfdatabase_username> <ctfdatabasename>
```

b) Reporting database:

```
createuser -p <reports_database_port> -P -S --createdb --no-createrole
<reports_database_username>
createuser -p <reports_database_port> -P -S --createdb --no-createrole
<reports_database_readonlyusername>
createdb -p <reports_database_port> -E UTF8 -O <reports database username>
<reports_database_name>
```

☞ **Note:**

- For the passwords, use the same passwords you recorded in the `site-options.conf` file.
- To specify a non-default port, add the -p option.

c) Restart PostgreSQL.

```
    exit
    /etc/init.d/postgresql-9.0 restart
```

Do this on `my.app.box`

14. Set up the initial site data.

```
    cd /opt/collabnet/teamforge-installer/6.1.1.0
    ./bootstrap-data.sh
```

15. Swap in the new Apache configuration file.

```
    cd /etc/httpd/conf
    mv httpd.conf httpd.conf_old
    cp httpd.conf.cn_new httpd.conf
    /etc/init.d/httpd start
```

Do this on the source control server. We'll call this `my.code.box`.

16. Install Red Hat Enterprise Linux 5.6 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 6.1.1* on page 96 for the full platform requirements.
- See *the Red Hat installation guide* for help.

☞   **Important:** Don't customize your installation. Select only the default packages list.

17. Check your basic networking setup.

See *Set up networking for your TeamForge box* on page 8 for details.

18. Download the TeamForge 6.1.1 installation repository from *open.collab.net*. Copy it to /etc/yum.repos.d/.

19. Install the TeamForge source control service.

```
   yum install teamforge-scm
```

☞   **Tip:** If the yum installer balks, you may have duplicate rpm packages.

    1. Get the `yum-utils` package, if it isn't already installed.

```
        yum install yum-utils
```

    2. Check for duplicate packages.

```
        package-cleanup --dupes
```

    3. Clean up the older packages, if any.

```
        package-cleanup --cleandupes
```

    4. Rerun the yum installer.

```
        yum install teamforge-scm
```

20. Rename the sample site configuration file from the installation package.

```
    cd /opt/collabnet/teamforge-installer/6.1.1.0
    cp conf/site-options-advanced.conf conf/site-options.conf
```

☞ **Note:** The files `site-options-small.conf`, `site-options-medium.conf` and `site-options-large.conf` contain options to tune the performance of the TeamForge site. To tune your site's performance, you can look through these files for the load specifications they are intended for, and use the appropriate one for your site's requirements.

21. Configure the `site-options.conf` file.

    a) Identify the boxes and the services running on them.

    ```
    HOST_localhost=subversion cvs
    DOMAIN_localhost=<myscmbox.domain.com>
    HOST_<myappbox.domain.com>=app
    HOST_<mydbbox.domain.com>=database
    ```

    b) Configure the database connection.

    ```
    DATABASE_TYPE=postgresql
    DATABASE_USERNAME=ctfuser
    DATABASE_PASSWORD=ctfpwd
    DATABASE_NAME=ctfdb
    ```

    c) Use the shared secret key, *SCM_DEFAULT_SHARED_SECRET*, from the `site-options.conf` file on the primary TeamForge server.

    d) If you don't specify a domain name, replace the HOST_localhost token with the hostname. Otherwise, ViewVC pages for Subversion and CVS repositories created after the site is up, will be rendered with a CSS error.

    e) Don't forget to save the file.

22. Run the installer.

    ```
    cd /opt/collabnet/teamforge-installer/6.1.1.0
    ./install.sh -r -I -V
    ```

23. Set up the initial site data.

    ```
    cd /opt/collabnet/teamforge-installer/6.1.1.0
    ./bootstrap-data.sh
    ```

24. If you are installing on a machine that is behind a proxy, unset the *http_proxy* variable.

    ```
    export http_proxy=
    ```

25. Swap in the new Apache configuration file.

    ```
    cd /etc/httpd/conf
    mv httpd.conf httpd.conf_old
    cp httpd.conf.cn_new httpd.conf
    /etc/init.d/httpd start
    ```

26. Start TeamForge.

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

Do this on `my.app.box`

27. Start TeamForge.

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

☞ **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 600 seconds, giving up now. Please check the log:
/opt/collabnet/teamforge/log/apps/service.log FAILED
```

This can safely be ignored.

Your installation is complete. Go to *Verify your TeamForge 6.1.1 installation* on page 40 for recommended steps to get started with your site.

Option 6: Distribute TeamForge 6.1.1 services using Oracle

Option 6 is like Option 5, but it uses an Oracle database instead of a PostgreSQL database.

For simplicity, in this example we will have the site database and the reporting database share port 1521.

☞ **Note:** If either of the remote machines (the data box or the source code box) is not under your direct control, check with the DBA or other person in charge of that resource to make sure you can carry out these instructions on that box.

Do this on the main TeamForge application server. We'll call this `my.app.box`.

1. Install Red Hat Enterprise Linux 5.6 and log in as root.

   - The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 6.1.1* on page 96 for the full platform requirements.
   - See *the Red Hat installation guide* for help.

   ☞ **Important:** Don't customize your installation. Select only the default packages list.

2. Check your basic networking setup.

   See *Set up networking for your TeamForge box* on page 8 for details.

3. Download the TeamForge 6.1.1 installation repository from *open.collab.net*. Copy it to `/etc/yum.repos.d/`.

4. Install the TeamForge application and reporting service.

   ```
   yum install teamforge-app teamforge-etl
   ```

   ☞ **Tip:** If the yum installer balks, you may have duplicate rpm packages.

   1. Get the `yum-utils` package, if it isn't already installed.

      ```
      yum install yum-utils
      ```

   2. Check for duplicate packages.

      ```
      package-cleanup --dupes
      ```

   3. Clean up the older packages, if any.

      ```
      package-cleanup --cleandupes
      ```

   4. Rerun the yum installer.

      ```
      yum install teamforge-app teamforge-etl
      ```

5. Rename the sample site configuration file from the installation package.

   ```
   cd /opt/collabnet/teamforge-installer/6.1.1.0
   ```

```
cp conf/site-options-advanced.conf conf/site-options.conf
```

☞ **Note:** The files `site-options-small.conf`, `site-options-medium.conf` and `site-options-large.conf` contain options to tune the performance of the TeamForge site. To tune your site's performance, you can look through these files for the load specifications they are intended for, and use the appropriate one for your site's requirements.

6. Configure the site-options.conf file.

```
vi conf/site-options.conf
```

a) Identify the boxes and the services running on them.

```
HOST_localhost=app etl
DOMAIN_localhost=<myappbox.domain.com>
HOST_<mydbbox.domain.com>=database datamart
HOST_<myscmbox.domain.com>=subversion cvs
```

b) Configure the database and the datamart.

☞ **Tip:** For more information about configuring variables, see *site-options.conf* on page 112.

```
DATABASE_TYPE=oracle
DATABASE_USERNAME=sitedatabaseusername
DATABASE_PASSWORD=sitedatabasepwd
DATABASE_NAME=sitedatabaseinstancename
DATABASE_MAX_POOL_SIZE=100
REPORTS_DATABASE_USERNAME=reportingdatabaseusername
REPORTS_DATABASE_PASSWORD=reportingdatabasepwd
REPORTS_DATABASE_NAME=reportingdatabaseinstancename
REPORTS_DATABASE_READ_ONLY_USER=reportingreadonlyusername
REPORTS_DATABASE_READ_ONLY_PASSWORD=reportingreadonlyuserpwd
ETL_SOAP_SHARED_SECRET=<arbitrary_string>
SOAP_ANONYMOUS_SHARED_SECRET=<arbitrary_string>
DATABASE_SERVICE_NAME=
REPORTS_DATABASE_SERVICE_NAME=
```

☞ **Tip:**

- The database name and user name values are arbitrary alphanumeric strings.
- To find the value for the *DATABASE_SERVICE_NAME* variable, log into your Oracle server and execute this command:

  ```
  su - oracle
  tnsping <database_name>
  ```

  In the output, find the value of *SERVICE_NAME*. Use that as the value of *DATABASE_SERVICE_NAME* in `site-options.conf`.

c) If you don't specify a domain name, replace the HOST_localhost token with the hostname. Otherwise, ViewVC pages for Subversion and CVS repositories created after the site is up, will be rendered with a CSS error.

d) ☞ **Important: It is mandatory that you include the *SCM_DEFAULT_SHARED_SECRET* token in the site-options.conf file of the primary TeamForge server, and give it a value of 16-24 characters.**

Remember that you need to use that same key in the external SCM integration server also.

e) Don't forget to save the file.

7. ☞ **Note:** Perform this step in case your Oracle server version is not 11.2.0.1.

Download the corresponding version of Oracle client from
http://www.oracle.com/technetwork/database/features/instant-client/index-097480.html and run the following
command:

```
yum localinstall <path to oracle client rpm>
```

8. Run the installer.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
./install.sh -r -I -V
```

9. Copy the Oracle datamart setup script from /opt/collabnet/teamforge/runtime/scripts to the /tmp
directory of my.data.box.

```
scp /opt/collabnet/teamforge/runtime/scripts/datamart-oracle-setup.sh
<username>@<my.data.box>:/tmp
```

Do this on the database/datamart server. We'll call this `my.data.box`.

10. Install Red Hat Enterprise Linux 5.6 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 6.1.1* on page
  96 for the full platform requirements.
- See *the Red Hat installation guide* for help.

☞ **Important:** Don't customize your installation. Select only the default packages list.

11. Install Oracle 11 (R1 and R2).

See *the Oracle wiki* for details.

☞ **Note:** Make sure your database uses UTF8 or AL32UTF8 encoding. This is needed to support users in Asian
languages. See *this Oracle knowledge base article.*

12. Put the Oracle datamart setup script where TeamForge can find it.

```
mkdir /u1
cp /tmp/datamart-oracle-setup.sh /u1
```

13. Log in as the Oracle user and create the site database user and permissions.

See *Set up an Oracle database* on page 74 for help.

14. Create the reporting user and schema.

```
cd /u1
sh datamart-oracle-setup.sh
```

☞ **Note:** Your responses to the script's prompts must match the values of the equivalent variables in the
site-options.conf file on my.app.box.

Do this on `my.app.box`

15. Set up the initial site data.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
./bootstrap-data.sh
```

16. If you are installing on a machine that is behind a proxy, unset the *http_proxy* variable.

```
export http_proxy=
```

17. Swap in the new Apache configuration file.

```
cd /etc/httpd/conf
mv httpd.conf httpd.conf_old
cp httpd.conf.cn_new httpd.conf
/etc/init.d/httpd start
```

Do this on the source control server. We'll call this `my.code.box`.

18. Install Red Hat Enterprise Linux 5.6 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 6.1.1* on page 96 for the full platform requirements.
- See *the Red Hat installation guide* for help.

☞ **Important:** Don't customize your installation. Select only the default packages list.

19. Check your basic networking setup.

See *Set up networking for your TeamForge box* on page 8 for details.

20. Download the TeamForge 6.1.1 installation repository from *open.collab.net*. Copy it to /etc/yum.repos.d/.

21. Install the TeamForge source control service.

```
yum install teamforge-scm
```

☞ **Tip:** If the yum installer balks, you may have duplicate rpm packages.

    1. Get the `yum-utils` package, if it isn't already installed.

```
yum install yum-utils
```

    2. Check for duplicate packages.

```
package-cleanup --dupes
```

    3. Clean up the older packages, if any.

```
package-cleanup --cleandupes
```

    4. Rerun the yum installer.

```
yum install teamforge-scm
```

22. Rename the sample site configuration file from the installation package.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
cp conf/site-options-advanced.conf conf/site-options.conf
```

☞ **Note:** The files `site-options-small.conf`, `site-options-medium.conf` and `site-options-large.conf` contain options to tune the performance of the TeamForge site. To tune your site's performance, you can look through these files for the load specifications they are intended for, and use the appropriate one for your site's requirements.

23. Configure the site-options.conf file.

    a) Identify the boxes and the services running on them.

```
HOST_localhost=subversion cvs
DOMAIN_localhost=<myscmbox.domain.com>
```

```
HOST_<myappbox.domain.com>=app
HOST_<mydbbox.domain.com>=database
```

b) Configure the database connection.

```
DATABASE_TYPE=oracle
DATABASE_USERNAME=ctfuser
DATABASE_PASSWORD=ctfpwd
DATABASE_NAME=ctfdb
```

c) Use the shared secret key, *SCM_DEFAULT_SHARED_SECRET*, from the `site-options.conf` file on the primary TeamForge server.

d) If you don't specify a domain name, replace the HOST_localhost token with the hostname. Otherwise, ViewVC pages for Subversion and CVS repositories created after the site is up, will be rendered with a CSS error.

e) Don't forget to save the file.

24. Run the installer.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
./install.sh -r -I -V
```

25. Set up the initial site data.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
./bootstrap-data.sh
```

26. If you are installing on a machine that is behind a proxy, unset the *http_proxy* variable.

```
export http_proxy=
```

27. Swap in the new Apache configuration file.

```
cd /etc/httpd/conf
mv httpd.conf httpd.conf_old
cp httpd.conf.cn_new httpd.conf
/etc/init.d/httpd start
```

28. Start TeamForge.

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

Do this on `my.app.box`

29. Start TeamForge.

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

> 👉 **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 600 seconds, giving up now. Please check the log:
> /opt/collabnet/teamforge/log/apps/service.log FAILED
> ```
>
> This can safely be ignored.

Your installation is complete. Go to *Verify your TeamForge 6.1.1 installation* on page 40 for recommended steps to get started with your site.

Verify your TeamForge 6.1.1 installation

Congratulations: you have just installed your TeamForge 6.1.1 site on Red Hat . Now you can apply some finishing touches and make sure everything is running smoothly.

1. Turn on SSL for your site by editing the relevant variables in the `site-options.conf` file.

   See *Set up SSL for your TeamForge site* on page 81 for details.

   ☞   **Note:** If SSL is enabled for any box belonging to your site, it must be enabled for all of them.

2. Log into your site as the administrator.

   The value of the *DOMAIN* variable in the `site-options.conf` file is the URL to log into.

3. Install your license key.

   See *Supply your TeamForge license key* on page 66.

4. Install a project template.

   TeamForge comes with a sample project template that showcases some of the platform's most interesting features. Site administrators and project managers can use this template to jump-start projects without a lot of manual setup steps. See *Install project templates* on page 78.

   ☞   **Note:** This procedure is recommended, but not required.

5. Create a sample project.

   See *Create a TeamForge project*.

6. Write a welcome message to your site's users.

   See *Create a site-wide broadcast*.

Now that you have successfully installed your TeamForge site in its basic configuration, you can use the instructions under *Maintain your CollabNet TeamForge 6.1.1 site on Red Hat* on page 66 to help keep your site going.

Uninstall TeamForge 6.1.1

To remove TeamForge completely, use the yum utility.

☞   **Important:** This procedure removes the TeamForge and all associated databases, including your site data. Be sure to back up any data you want to keep.

1. Stop TeamForge.

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

2. Run yum to remove TeamForge.

   ```
   yum erase TeamForge-installer
   ```

For every box in a multi-box site, use the same steps to uninstall.

Upgrade to TeamForge 6.1.1 on Red Hat

You can upgrade to TeamForge 6.1.1 from TeamForge 6.1. You can upgrade on the same box where your current TeamForge site is running, or you can take this opportunity to move your site to a new box. Choose the instructions that fit your case.

☞ **Note:** Due to the Lucene upgrade for search functionality, upgrading to TeamForge 6.1.1 requires a complete re-index of the site. This could take several hours, and the index data could double in size.

Plan your upgrade to TeamForge 6.1.1

As the first step in upgrading to TeamForge 6.1.1, consider some key questions that will affect how your new site works.

1. Where is everything?

    TeamForge consists of five interrelated services that can run on separate hardware or share one or more boxes in various configurations. If you aren't the person who first installed your current TeamForge site (or maybe even if you are), it's essential to catalog the hosts where your services are running and to know what configuration has been applied to them.

2. Provide site-wide reporting?

    TeamForge 6.1.x enables site administrators to track user logins for their site. More reporting options are coming.

    If your users want this feature, you'll have to turn on a service known as extract-transform-load (ETL), during the upgrade process. You'll also have to configure a new database called the datamart. (These elements are off by default.)

    After your site is upgraded, you can also choose to move these new services off to one or more separate servers, the same as you can do with the existing TeamForge services.

3. "Dedicated" or "advanced?"

    The type of TeamForge installation you have makes a difference for how you upgrade and patch the site. If your site is a dedicated site, you'll be able to skip some of the steps outlined here. If you don't know whether your site was originally installed as dedicated or advanced, here's how you can find out: *Is my TeamForge site "dedicated" or "advanced?"* on page 59

4. Branding changes?

    Every release of TeamForge can bring changes to the look and feel of the product. TeamForge 6.1.1 is no exception. If you have edited files in your site's branding repository (that's how you customize the look and feel of the product), you must download the new branding package and check into your branding repository the new versions of any files you have edited. See *Customize anything on your site* for instructions.

5. Special database settings?

    The efficiency of your database can have an impact on your users' perception of the site's usability. If your site uses a PostgreSQL database (which is the default), you may want to consider tuning it to fit your specific circumstances. The default settings are intended for a small-to-medium site running on a single server. See *What are the right PostgreSQL settings for my site?* on page 165 for recommendations from CollabNet's performance team on optimizing PostgreSQL for different conditions.

6. Upgraded JDK

    TeamForge 6.1.1 uses JDK 1.6.0_26. If you are upgrading on the same box, and that box has a JDK at an earlier version, the upgrade utility upgrades the JDK. However, you'll still need to edit the *JAVA_HOME* variable in your `site-options.conf` file to reflect the new JDK version. See *Upgrade a TeamForge site to 6.1.1 on the same hardware* on page 48.

Upgrade a dedicated CollabNet TeamForge site to 6.1.1

Upgrading to TeamForge 6.1.1 on a dedicated site is easy and fun.

Upgrade a dedicated CollabNet TeamForge site to 6.1.1 on the same box

To upgrade to TeamForge 6.1.1, you install the new site and convert your existing site's data.

For this procedure, we'll assume that you are upgrading on the same box where your existing TeamForge site is running. It's also possible to simultaneously upgrade and move your site to new hardware, but since we are working with a dedicated installation, the priority here is to keep things as simple and quick as possible. See *Should I upgrade to TeamForge 6.1.1 on a new box?* on page 164 for some background information.

☞ **Tip:** If there is any doubt about what kind of site you are working with, see *Is my TeamForge site "dedicated" or "advanced?"* on page 59

1. Back up your site data.

   a) Make a dump file of your site database.

   (This may be the same as your TeamForge application server or a separate box.)

   You have to do a PostgreSQL dump because we are upgrading the PostgreSQL application as part of this upgrade.

   ☞ **Note:** These commands are for a PostgreSQL database, which is the default. If your site uses an Oracle database, follow the *Oracle backup procedure* instead.

   ```
   su - postgres
   /usr/bin/pg_dumpall > /var/lib/pgsql/backups/teamforge_data_backup.dmp
   exit
   mkdir /tmp/backup_dir
   cp /var/lib/pgsql/backups/teamforge_data_backup.dmp /tmp/backup_dir/
   ```

   ☞ **Note:** If your reporting database is running on a separate port, back up your reporting database too:

   ```
   /usr/bin/pg_dumpall -p <reports_database_port> >
   /var/lib/pgsql/backups/teamforge_reporting_data_backup.dmp
   ```

   b) Make an archive file with the following data directories:

Directory	Contents
/opt/collabnet/teamforge/var	User-created data, such as artifact attachments
/svnroot	Subversion source code repositories
/sf-svnroot	Subversion repository for branding data
/cvsroot	CVS source code repositories (not present on all sites)

   ```
   cp -Rpf /svnroot /sf-svnroot /cvsroot /opt/collabnet/teamforge/var
   /tmp/backup_dir
   cd /tmp
   tar czvf 61backup.tgz backup_dir
   ```

   c) Back up your SSH keys, if any.

   d) Back up your SSL certificates and keys, if any.

2. Stop TeamForge.

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

3. Uninstall the PostgreSQL support packages to clear the way for TeamForge to install PostgreSQL 9.0. (These packages are installed with the OS by default.)

```
yum erase postgresql-libs postgresql-docs
```

4. Move the repositories from any previous installs out of the way.

```
mv /etc/yum.repos.d/collabnet-6.1.0.0.repo
/etc/yum.repos.d/collabnet-6.1.0.0.repo.cn_backup
```

5. Download the TeamForge 6.1.1 installation repository from *open.collab.net*. Copy it to `/etc/yum.repos.d/`.

6. Install the TeamForge application.

```
yum install teamforge-app teamforge-scm teamforge-database teamforge-etl
```

> ☞ **Tip:** If the yum installer balks, you may have duplicate rpm packages.
>
>   1. Get the `yum-utils` package, if it isn't already installed.
>
>      ```
>      yum install yum-utils
>      ```
>
>   2. Check for duplicate packages.
>
>      ```
>      package-cleanup --dupes
>      ```
>
>   3. Clean up the older packages, if any.
>
>      ```
>      package-cleanup --cleandupes
>      ```
>
>   4. Rerun the yum installer.
>
>      ```
>      yum install teamforge [options]
>      ```

7. Make sure your Java settings are right for TeamForge 6.1.1.

   a) Check the *JAVA_HOME* variable in `/opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf` to make sure TeamForge is using the right JDK.

   ```
   vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
   ```

   The JAVA_HOME variable should look like this:

   ```
   JAVA_HOME=/usr/java/jdk1.6.0_26
   ```

   b) In the *JBOSS_JAVA_OPTS* variable, increase the MaxPermSize value to 512.

8. In the `site-options.conf` file, make sure you have the following:

   a) If you don't specify a domain name, replace the HOST_localhost token with the hostname. Otherwise, ViewVC pages for Subversion and CVS repositories created after the site is up, will be rendered with a CSS error.

   b) ☞ **Important: It is mandatory that you include the *SCM_DEFAULT_SHARED_SECRET* token in the site-options.conf file of the primary TeamForge server, and give it a value of 16-24 characters.**

   Remember that you need to use that same key in the external SCM integration server also.

9. Run the installer.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
./install.sh -r -I -V
```

10. Bring your site data back.

```
su - postgres
/usr/bin/psql < /var/lib/pgsql/backups/teamforge_data_backup.dmp
exit
```

☞ **Note:** If your reporting database is running on a separate port, restore that data too:

```
su - postgres -c "/usr/bin/psql -p <reports_database_port> <
/var/lib/pgsql/backups/teamforge_reporting_data_backup.dmp"
```

11. Update the file permissions on your site's data.

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

☞ **Note:** This process can take a long time for a site with a lot of data.

12. Convert your site data to work with TeamForge 6.1.1.

☞ **Tip:** Before you kick off the data migration, use the `/etc/init.d/collabnet status` command to make sure Jboss and Tomcat are stopped.

```
/opt/collabnet/teamforge/runtime/scripts/migrate.py
```

The `migrate.py` script locates the existing site data and modifies it as needed.

This includes configuration data for LDAP and the James mail server. Any modifications that you have applied to these components on your old site are reproduced on your upgraded TeamForge 6.1.1 site.

13. Start TeamForge.

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

14. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 6.1.1.
   a) Make sure your users can still access their source control services.

   See *Synchronize TeamForge source control integrations* on page 87.
   b) If you are bringing SSL certificates and keys from your old site, make sure their locations match the paths specified by the *SSL_CERT_FILE* and *SSL_KEY_FILE* variables in `site-options.conf`. See *Set up SSL for your TeamForge site* on page 81.
   c) Log into your site as the administrator.
   d) Reboot the server and make sure all services come up automatically at startup.
   e) Rebuild your site's search index so that users get up-to-date search results.

   See *Rebuild TeamForge search indexes* on page 74 for details.
   f) If your site has custom branding, verify that your branding changes still work as intended.

   See *Customize anything on your site*.
   g) Let your site's users know they've been upgraded.

   See *Create a site-wide broadcast*.

☞ **Important:** Do not delete the `teamforge-installer/6.1.1.0` directory. You will need it for future maintenance and upgrades.

☞ **Note:** After the upgrade, it takes some time for the publishing repositories to get created for projects imported from other TeamForge sites.

Upgrade a dedicated site to TeamForge 6.1.1 on new hardware
To upgrade to TeamForge 6.1.1, set up a new box, then bring over your old site's data and convert it.

TeamForge 6.1.1 runs on Red Hat 5.6. To take advantage of the new operating system, you should install a fresh box and move your TeamForge site to it.

If necessary, however, TeamForge 6.1.1 can still run on Red Hat 5.5. If you need to run your new site on the same box that your existing site is running on, see

On the existing TeamForge application box, log in as root and take these steps:

1. Stop the TeamForge application server and the Apache server, if they are running.

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

2. Back up your site data.

a) Make a dump file of your site database.

(This may be the same as your TeamForge application server or a separate box.)

You have to do a PostgreSQL dump because we are upgrading the PostgreSQL application as part of this upgrade.

☞ **Note:** These commands are for a PostgreSQL database, which is the default. If your site uses an Oracle database, follow the *Oracle backup procedure* instead.

```
su - postgres
/usr/bin/pg_dumpall > /var/lib/pgsql/backups/teamforge_data_backup.dmp
exit
mkdir /tmp/backup_dir
cp /var/lib/pgsql/backups/teamforge_data_backup.dmp /tmp/backup_dir/
```

☞ **Note:** If your reporting database is running on a separate port, back up your reporting database too:

```
/usr/bin/pg_dumpall -p <reports_database_port> >
/var/lib/pgsql/backups/teamforge_reporting_data_backup.dmp
```

b) Make an archive file with the following data directories:

Directory	Contents
/opt/collabnet/teamforge/var	User-created data, such as artifact attachments
/svnroot	Subversion source code repositories
/sf-svnroot	Subversion repository for branding data
/cvsroot	CVS source code repositories (not present on all sites)

```
cp -Rpf /svnroot /sf-svnroot /cvsroot /opt/collabnet/teamforge/var
/tmp/backup_dir
cd /tmp
tar czvf 61backup.tgz backup_dir
```

c) Back up your SSH keys, if any.

d) Back up your SSL certificates and keys, if any.

3. Stop PostgreSQL.

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

4. Copy the file system data to the new box.

```
scp /tmp/61backup.tgz username@newbox:/tmp
```

5. Copy the master configuration file from the old server to the same location on the new server.

```
scp /opt/collabnet/teamforge-installer/6.1.0.0/conf/site-options.conf
username@newbox:/tmp
```

☞ **Tip:** `scp` is just an example. Use whatever file transfer method you prefer.

Do this on the new TeamForge 6.1.1 box:

6. Install Red Hat Enterprise Linux 5.6 and log in as root.

   • The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 6.1.1* on page 96 for the full platform requirements.
   • See *the Red Hat installation guide* for help.

   ☞ **Important:** Don't customize your installation. Select only the default packages list.

7. Download the TeamForge 6.1.1 installation repository from *open.collab.net*. Copy it to `/etc/yum.repos.d/`.

8. Install the TeamForge application.

```
yum install teamforge-app teamforge-scm teamforge-database teamforge-etl
```

   ☞ **Tip:** If the yum installer balks, you may have duplicate rpm packages.

   1. Get the `yum-utils` package, if it isn't already installed.

```
yum install yum-utils
```

   2. Check for duplicate packages.

```
package-cleanup --dupes
```

   3. Clean up the older packages, if any.

```
package-cleanup --cleandupes
```

   4. Rerun the yum installer.

```
yum install teamforge [options]
```

9. Put the `site-options.conf` file where the installer can find it.

```
cp /tmp/site-options.conf /opt/collabnet/teamforge-installer/6.1.1.0/conf
```

10. Log in as root and unpack the file system data.

```
cd /tmp
tar xzvf 61backup.tgz
```

11. Open the `site-options.conf` file.
    This is the master configuration file that controls your TeamForge site.

```
vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
```

☞   **Note:** `vi` is an example. Any *nix text editor will work.

12. Update the host name and domain name, if necessary.

```
HOST_my.host.name=app database subversion cvs etl datamart
DOMAIN_my.host.name=my.domain.name
```

13. Make sure your Java settings are right for TeamForge 6.1.1.

a) Check the *JAVA_HOME* variable in
`/opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf` to make sure
TeamForge is using the right JDK.

```
vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
```

The JAVA_HOME variable should look like this:

```
JAVA_HOME=/usr/java/jdk1.6.0_26
```

b) In the *JBOSS_JAVA_OPTS* variable, increase the MaxPermSize value to 512.

14. If you don't specify a domain name, replace the HOST_localhost token with the hostname. Otherwise, ViewVC
pages for Subversion and CVS repositories created after the site is up, will be rendered with a CSS error.

15. ☞   **Important: It is mandatory that you include the *SCM_DEFAULT_SHARED_SECRET* token in the
site-options.conf file of the primary TeamForge server, and give it a value of 16-24 characters.**

Remember that you need to use that same key in the external SCM integration server also.

16. Review the variables you've changed, then save the `site-options.conf` file.

17. Run the installer.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
./install.sh -r -I -V
```

18. Bring your site data back.

a) Reload the PostgreSQL data.

```
su - postgres
/usr/bin/psql < /tmp/backup_dir/teamforge_data_backup.dmp
exit
```

☞   **Note:** If your reporting database is running on a separate port, restore that data too:

```
su - postgres -c "/usr/bin/psql -p <reports_database_port> <
/tmp/backup_dir/teamforge_reporting_data_backup.dmp"
```

b) Reload `svnroot`, `sf-svnroot`, `cvsroot`, and `var`.

19. Update the file permissions on your site's data.

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

☞   **Note:** This process can take a long time for a site with a lot of data.

20. Convert your site data to work with TeamForge 6.1.1.

☞   **Tip:** Before you kick off the data migration, use the `/etc/init.d/collabnet status` command
to make sure Jboss and Tomcat are stopped.

```
/opt/collabnet/teamforge/runtime/scripts/migrate.py
```

The `migrate.py` script locates the existing site data and modifies it as needed.

This includes configuration data for LDAP and the James mail server. Any modifications that you have applied to these components on your old site are reproduced on your upgraded TeamForge 6.1.1 site.

21. Start TeamForge.

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

22. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 6.1.1.

   a) Make sure your users can still access their source control services.

      See *Synchronize TeamForge source control integrations* on page 87.

   b) If you are bringing SSL certificates and keys from your old site, make sure their locations match the paths specified by the *SSL_CERT_FILE* and *SSL_KEY_FILE* variables in `site-options.conf`. See *Set up SSL for your TeamForge site* on page 81.

   c) Log into your site as the administrator.

   d) Reboot the server and make sure all services come up automatically at startup.

   e) Rebuild your site's search index so that users get up-to-date search results.

      See *Rebuild TeamForge search indexes* on page 74 for details.

   f) If your site has custom branding, verify that your branding changes still work as intended.

      See *Customize anything on your site*.

   g) Let your site's users know they've been upgraded.

      See *Create a site-wide broadcast*.

☞   **Important:** Do not delete the `teamforge-installer/6.1.1.0` directory. You will need it for future maintenance and upgrades.

☞   **Note:** After the upgrade, it takes some time for the publishing repositories to get created for projects imported from other TeamForge sites.

Upgrade an advanced TeamForge site to TeamForge 6.1.1

Upgrading to TeamForge 6.1.1 on an advanced site can be complicated but you get more flexibility and control.

If there is any doubt about what kind of site you are working with, see *Is my TeamForge site "dedicated" or "advanced?"* on page 59

Upgrade a TeamForge site to 6.1.1 on the same hardware

To upgrade to TeamForge 6.1.1 without changing the box where your TeamForge 6.1 site is running, you have to install the new TeamForge site and migrate the site data.

There are a few critical factors to keep in mind when upgrading to TeamForge 6.1.1.

- For this procedure, we'll assume that you are upgrading on the same box where your existing TeamForge site is running. It's also possible to simultaneously upgrade and move your site to new hardware. To do that, see *Upgrade to TeamForge 6.1.1 on new hardware* on page 53.

- This procedure is meant for sites that were installed in "advanced" mode. If your existing site was installed in "dedicated" mode, use *Upgrade a dedicated CollabNet TeamForge site to 6.1.1 on the same box* on page 42 instead. If there is any doubt about what kind of site you are working with, see *Is my TeamForge site "dedicated" or "advanced?"* on page 59

1. Back up your site data.

   a) Make a dump file of your site database.

(This may be the same as your TeamForge application server or a separate box.)

You have to do a PostgreSQL dump because we are upgrading the PostgreSQL application as part of this upgrade.

> ☞ **Note:** These commands are for a PostgreSQL database, which is the default. If your site uses an Oracle database, follow the *Oracle backup procedure* instead.

```
su - postgres
/usr/bin/pg_dumpall > /var/lib/pgsql/backups/teamforge_data_backup.dmp
exit
mkdir /tmp/backup_dir
cp /var/lib/pgsql/backups/teamforge_data_backup.dmp /tmp/backup_dir/
```

> ☞ **Note:** If your reporting database is running on a separate port, back up your reporting database too:
>
> ```
> /usr/bin/pg_dumpall -p <reports_database_port> >
> /var/lib/pgsql/backups/teamforge_reporting_data_backup.dmp
> ```

b) Make an archive file with the following data directories:

Directory	Contents
/opt/collabnet/teamforge/var	User-created data, such as artifact attachments
/svnroot	Subversion source code repositories
/sf-svnroot	Subversion repository for branding data
/cvsroot	CVS source code repositories (not present on all sites)

```
cp -Rpf /svnroot /sf-svnroot /cvsroot /opt/collabnet/teamforge/var
/tmp/backup_dir
cd /tmp
tar czvf 61backup.tgz backup_dir
```

c) Back up your SSH keys, if any.

d) Back up your SSL certificates and keys, if any.

2. Stop TeamForge.

```
/etc/init.d/httpd stop
/etc/init.d/postgresql-9.0 stop
/etc/init.d/collabnet stop
```

3. If the database is running on a separate box, go to that box and move the old PostgreSQL control script out of the way.

```
mv /etc/init.d/postgresql_ctf /etc/init.d/postgresql_old
```

4. If the reporting database is running on a separate box, go to that box and move the old datamart PostgreSQL control script out of the way.

```
mv /etc/init.d/postgresql_reports /etc/init.d/postgresql_reports_old
```

5. Uninstall the PostgreSQL support packages to clear the way for TeamForge to install PostgreSQL 9.0.

(These packages are installed with the OS by default.)

```
yum erase postgresql-libs postgresql-docs
```

6. Move the repositories from any previous installs out of the way.

```
mv /etc/yum.repos.d/collabnet-6.1.0.0.repo
/etc/yum.repos.d/collabnet-6.1.0.0.repo.cn_backup
```

7. Download the TeamForge 6.1.1 installation repository from *open.collab.net*. Copy it to `/etc/yum.repos.d/`.

8. Install the TeamForge application.

```
yum install teamforge-app teamforge-scm teamforge-database teamforge-etl
```

☞ **Tip:** If the yum installer balks, you may have duplicate rpm packages.

1. Get the `yum-utils` package, if it isn't already installed.

```
yum install yum-utils
```

2. Check for duplicate packages.

```
package-cleanup --dupes
```

3. Clean up the older packages, if any.

```
package-cleanup --cleandupes
```

4. Rerun the yum installer.

```
yum install teamforge [options]
```

9. Make sure your Java settings are right for TeamForge 6.1.1.
   a) Check the *JAVA_HOME* variable in
   `/opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf` to make sure
   TeamForge is using the right JDK.

```
vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
```

The JAVA_HOME variable should look like this:

```
JAVA_HOME=/usr/java/jdk1.6.0_26
```

   b) In the *JBOSS_JAVA_OPTS* variable, increase the MaxPermSize value to 512.

10. If you don't specify a domain name, replace the HOST_localhost token with the hostname. Otherwise, ViewVC
    pages for Subversion and CVS repositories created after the site is up, will be rendered with a CSS error.

11. ☞ **Important: It is mandatory that you include the *SCM_DEFAULT_SHARED_SECRET* token in the
    site-options.conf file of the primary TeamForge server, and give it a value of 16-24 characters.**

Remember that you need to use that same key in the external SCM integration server also.

12. Run the installer.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
./install.sh -r -I -V
```

13. Set up the site database.

```
su - postgres
/usr/pgsql-9.0/bin/initdb -D /var/lib/pgsql/9.0/data/
cp /var/lib/pgsql/data/postgresql.conf
/var/lib/pgsql/9.0/data/postgresql.conf
```

```
cp /var/lib/pgsql/data/pg_hba.conf /var/lib/pgsql/9.0/data/pg_hba.conf
exit
/etc/init.d/postgresql-9.0 start
```

14. If the reporting database (datamart) runs on a different port, you need to set it up.

a) Point the database to the local machine.

```
su - postgres
/usr/pgsql-9.0/bin/initdb -D /var/lib/pgsql/9.0/reports
vi /var/lib/pgsql/9.0/reports/postgresql.conf
```

```
    listen_addresses = '127.0.0.1,<IP address of datamart box>'
    port=5632
```

b) Configure access for the datamart.

```
vi /var/lib/pgsql/9.0/reports/pg_hba.conf
```

```
local   all                             all
                            trust
# IPv4 local connections:
host    all                             all
   127.0.0.1/32            trust
# IPv6 local connections:
#host   all                             all
   ::1/128                 trust
host    <REPORTS_DATABASE_NAME>         <REPORTS_DATABASE_USERNAME>
  <IP address of my.app.box>/32   md5
host    <REPORTS_DATABASE_NAME>         <REPORTS_DATABASE_READ_ONLY_USER>
  <IP address of my.app.box>/32   md5
```

```
exit
```

c) Copy the datamart control script from `/opt/collabnet/teamforge/runtime/scripts` on `my.app.box` to the `/tmp` directory of `my.datamart.box`.

```
mv /etc/init.d/postgresql-9.0 /etc/init.d/postgresql-9.0_orig
cp /tmp/postgresql_reports /etc/init.d
```

☞ **Note:** The `postgresql_reports` script is the one to use only if the datamart is running on a different server from the database, as it is in this example. In all other cases, use the `postgresql` script.

d) Start the reporting database service.

```
/etc/init.d/postgresql-9.0_reports start
```

15. Bring your site data back.

```
su - postgres
/usr/bin/psql < /var/lib/pgsql/backups/teamforge_data_backup.dmp
exit
```

☞ **Note:** If your reporting database is running on a separate port, restore that data too:

```
su - postgres -c "/usr/bin/psql -p <reports_database_port> <
/var/lib/pgsql/backups/teamforge_reporting_data_backup.dmp"
```

16. Update the file permissions on your site's data.

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

☞ **Note:** This process can take a long time for a site with a lot of data.

17. Convert your site data to work with TeamForge 6.1.1.

☞ **Tip:** Before you kick off the data migration, use the /etc/init.d/collabnet status command to make sure Jboss and Tomcat are stopped.

```
/opt/collabnet/teamforge/runtime/scripts/migrate.py
```

The migrate.py script locates the existing site data and modifies it as needed.

This includes configuration data for LDAP and the James mail server. Any modifications that you have applied to these components on your old site are reproduced on your upgraded TeamForge 6.1.1 site.

18. Swap in the new Apache configuration file.

```
cd /etc/httpd/conf
mv httpd.conf httpd.conf_old
cp httpd.conf.cn_new httpd.conf
/etc/init.d/httpd start
```

19. Start TeamForge.

Start PostgreSQL if it is not running.

```
/etc/init.d/postgresql-9.0 start
```

Start the CollabNet services.

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

20. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 6.1.1.
   a) Make sure your users can still access their source control services.

   See *Synchronize TeamForge source control integrations* on page 87.
   b) If you are bringing SSL certificates and keys from your old site, make sure their locations match the paths specified by the *SSL_CERT_FILE* and *SSL_KEY_FILE* variables in site-options.conf. See *Set up SSL for your TeamForge site* on page 81.
   c) Log into your site as the administrator.
   d) Reboot the server and make sure all services come up automatically at startup.
   e) Rebuild your site's search index so that users get up-to-date search results.

   See *Rebuild TeamForge search indexes* on page 74 for details.
   f) If your site has custom branding, verify that your branding changes still work as intended.

   See *Customize anything on your site*.
   g) Let your site's users know they've been upgraded.

   See *Create a site-wide broadcast*.

☞ **Important:** Do not delete the teamforge-installer/6.1.1.0 directory. You will need it for future maintenance and upgrades.

☞ **Note:** After the upgrade, it takes some time for the publishing repositories to get created for projects imported from other TeamForge sites.

Upgrade to TeamForge 6.1.1 on new hardware

To upgrade to TeamForge 6.1.1, set up a new box, then bring over your old site's data and convert it.

TeamForge 6.1.1 runs on Red Hat Enterprise Linux5.6. To take advantage of the new operating system, you should install a fresh box and move your TeamForge 6.1 site to it.

If necessary, however, TeamForge 6.1.1 can still run on Red Hat 5.5. If you need to run your new site on the same box that your existing site is running on, see *Upgrade a TeamForge site to 6.1.1 on the same hardware* on page 48.

☞ **Note:** This procedure assumes that you are running your new site-wide reporting database on the same box where your TeamForge application and database are running, and that the reporting database shares the same port as the site database.

You don't have to set it up exactly like this.

- You can run the reporting database on a separate box. For an example of that, see *Option 3: Install TeamForge 6.1.1 with reporting data on a separate box* on page 20.
- You can run the reporting database on the same box as the site database, but with a separate port. For an example, see *Option 5: Distribute TeamForge 6.1.1 services a different way* on page 29

Take these steps on the new TeamForge 6.1.1 box:

1. Install Red Hat Enterprise Linux 5.6 and log in as root.

    - The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 6.1.1* on page 96 for the full platform requirements.
    - See *the Red Hat installation guide* for help.

    ☞ **Important:** Don't customize your installation. Select only the default packages list.

2. Check your basic networking setup.

    See *Set up networking for your TeamForge box* for details.

3. Download the TeamForge 6.1.1 installation repository from *open.collab.net*. Copy it to `/etc/yum.repos.d/`.

On the TeamForge 6.1 box, log in as root and take these steps:

4. Stop the TeamForge application server and the Apache server, if they are running.

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

5. Back up your site data.

    a) Make a dump file of your site database.

       (This may be the same as your TeamForge application server or a separate box.)

       You have to do a PostgreSQL dump because we are upgrading the PostgreSQL application as part of this upgrade.

       ☞ **Note:** These commands are for a PostgreSQL database, which is the default. If your site uses an Oracle database, follow the *Oracle backup procedure* instead.

       ```
       su - postgres
       /usr/bin/pg_dumpall > /var/lib/pgsql/backups/teamforge_data_backup.dmp
       exit
       mkdir /tmp/backup_dir
       cp /var/lib/pgsql/backups/teamforge_data_backup.dmp /tmp/backup_dir/
       ```

> 👉 **Note:** If your reporting database is running on a separate port, back up your reporting database too:
>
> ```
> /usr/bin/pg_dumpall -p <reports_database_port> >
> /var/lib/pgsql/backups/teamforge_reporting_data_backup.dmp
> ```

b) Make an archive file with the following data directories:

Directory	Contents
/opt/collabnet/teamforge/var	User-created data, such as artifact attachments
/svnroot	Subversion source code repositories
/sf-svnroot	Subversion repository for branding data
/cvsroot	CVS source code repositories (not present on all sites)

```
cp -Rpf /svnroot /sf-svnroot /cvsroot /opt/collabnet/teamforge/var
/tmp/backup_dir
cd /tmp
tar czvf 61backup.tgz backup_dir
```

c) Back up your SSH keys, if any.
d) Back up your SSL certificates and keys, if any.

6. Stop PostgreSQL.

```
sudo /etc/init.d/postgresql stop
```

7. If the database is running on a separate box, go to that box and move the old PostgreSQL control script out of the way.

```
mv /etc/init.d/postgresql_ctf /etc/init.d/postgresql_old
```

8. If the reporting database is running on a separate box, go to that box and move the old datamart PostgreSQL control script out of the way.

```
mv /etc/init.d/postgresql_reports /etc/init.d/postgresql_reports_old
```

9. Copy the file system data to the new box.

```
scp /tmp/61backup.tgz username@newbox:/tmp
```

10. Copy the master configuration file from the old server to the same location on the new server.

```
scp /opt/collabnet/teamforge-installer/6.1.0.0/conf/site-options.conf
username@newbox:/tmp
```

> 👉 **Tip:** scp is just an example. Use whatever file transfer method you prefer.

Do this on the new TeamForge 6.1.1 box:

11. Log in as root and unpack the file system data.

```
cd /tmp
tar xzvf 61backup.tgz
```

12. Install the TeamForge application.

```
yum install teamforge-app teamforge-scm teamforge-database teamforge-etl
```

☞ **Tip:** If the yum installer balks, you may have duplicate rpm packages.

1. Get the `yum-utils` package, if it isn't already installed.

   ```
   yum install yum-utils
   ```

2. Check for duplicate packages.

   ```
   package-cleanup --dupes
   ```

3. Clean up the older packages, if any.

   ```
   package-cleanup --cleandupes
   ```

4. Rerun the yum installer.

   ```
   yum install teamforge [options]
   ```

13. Put the `site-options.conf` file where the installer can find it.

```
cp /tmp/site-options.conf /opt/collabnet/teamforge-installer/6.1.1.0/conf
```

14. Open the `site-options.conf` file.

This is the master configuration file that controls your TeamForge site.

```
vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
```

☞ **Note:** `vi` is an example. Any *nix text editor will work.

15. Update the host name and domain name, if necessary.

```
HOST_my.host.name=app database subversion cvs etl datamart
DOMAIN_my.host.name=my.domain.name
```

16. Make sure your Java settings are right for TeamForge 6.1.1.

   a) Check the *JAVA_HOME* variable in
   `/opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf` to make sure
   TeamForge is using the right JDK.

   ```
   vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
   ```

   The JAVA_HOME variable should look like this:

   ```
   JAVA_HOME=/usr/java/jdk1.6.0_26
   ```

   b) In the *JBOSS_JAVA_OPTS* variable, increase the MaxPermSize value to 512.

17. If you don't specify a domain name, replace the HOST_localhost token with the hostname. Otherwise, ViewVC
pages for Subversion and CVS repositories created after the site is up, will be rendered with a CSS error.

18. ☞ **Important: It is mandatory that you include the *SCM_DEFAULT_SHARED_SECRET* token in the
site-options.conf file of the primary TeamForge server, and give it a value of 16-24 characters.**

Remember that you need to use that same key in the external SCM integration server also.

19. Review the variables you've changed, then save the `site-options.conf` file.

20. Run the installer.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
```

```
./install.sh -r -I -V
```

21. Set up the site database.

a) Point the database to the local machine.

```
su - postgres
/usr/pgsql-9.0/bin/initdb -D /var/lib/pgsql/9.0/data
vi /var/lib/pgsql/9.0/data/postgresql.conf
```

```
listen_addresses = '127.0.0.1,<IP address of database box>'
```

b) Configure database access for the site database.

```
vi /var/lib/pgsql/9.0/data/pg_hba.conf
```

```
local  all                        all
                        trust
# IPv4 local connections:
host    all                        all
127.0.0.1/32                   trust
# IPv6 local connections:
#host   all                        all                        ::1/128
                    trust
host    <DATABASE_NAME>            <DATABASE_USERNAME                <IP
address of app box>/32   md5
host    <REPORTS_DATABASE_NAME> <REPORTS_DATABASE_USERNAME>          <IP
address of app box>/32   md5
host    <REPORTS_DATABASE_NAME> <REPORTS_DATABASE_READ_ONLY_USERNAME> <IP
address of app box>/32   md5
```

c) Exit and restart PostgreSQL.

```
exit
/etc/init.d/postgresql-9.0 start
```

22. Bring your site data back.

a) Reload the PostgreSQL data.

```
su - postgres
/usr/bin/psql < /tmp/backup_dir/teamforge_data_backup.dmp
exit
```

> 👉 **Note:** If your reporting database is running on a separate port, restore that data too:
>
> ```
> su - postgres -c "/usr/bin/psql -p <reports_database_port> <
> /tmp/backup_dir/teamforge_reporting_data_backup.dmp"
> ```

b) Reload `svnroot`, `sf-svnroot`, `cvsroot`, and `var`.

23. Update the file permissions on your site's data.

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

> 👉 **Note:** This process can take a long time for a site with a lot of data.

24. Convert your site data to work with TeamForge 6.1.1.

> 👉 **Tip:** Before you kick off the data migration, use the `/etc/init.d/collabnet status` command to make sure Jboss and Tomcat are stopped.

```
/opt/collabnet/teamforge/runtime/scripts/migrate.py
```

The `migrate.py` script locates the existing site data and modifies it as needed.

This includes configuration data for LDAP and the James mail server. Any modifications that you have applied to these components on your old site are reproduced on your upgraded TeamForge 6.1.1 site.

25. Swap in the new Apache configuration file.

```
cd /etc/httpd/conf
mv httpd.conf httpd.conf_old
cp httpd.conf.cn_new httpd.conf
/etc/init.d/httpd start
```

26. Start TeamForge.

Start PostgreSQL if it is not running.

```
/etc/init.d/postgresql-9.0 start
```

Start CollabNet services.

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

27. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 6.1.1.

a) Make sure your users can still access their source control services.

See *Synchronize TeamForge source control integrations* on page 87.

b) If you are bringing SSL certificates and keys from your old site, make sure their locations match the paths specified by the *SSL_CERT_FILE* and *SSL_KEY_FILE* variables in `site-options.conf`. See *Set up SSL for your TeamForge site* on page 81.

c) Log into your site as the administrator.

d) Reboot the server and make sure all services come up automatically at startup.

e) Rebuild your site's search index so that users get up-to-date search results.

See *Rebuild TeamForge search indexes* on page 74 for details.

f) If your site has custom branding, verify that your branding changes still work as intended.

See *Customize anything on your site*.

g) Let your site's users know they've been upgraded.

See *Create a site-wide broadcast*.

☞   **Important:** Do not delete the `teamforge-installer/6.1.1.0` directory. You will need it for future maintenance and upgrades.

☞   **Note:** After the upgrade, it takes some time for the publishing repositories to get created for projects imported from other TeamForge sites.

Upgrade a stand-alone TeamForge source control box on the same hardware
If your TeamForge setup includes source control running on its own box, you'll have to upgrade that box as well as the main TeamForge application box.

For example, if you used *these install instructions* to install TeamForge 6.1.1, then this procedure is for you.

☞   **Note:** This procedure is meant for sites that were installed in "advanced" mode. If your existing site was installed in "dedicated" mode, use *Upgrade a dedicated CollabNet TeamForge site to 6.1.1 on the same box* on page 42 instead. If there is any doubt about what kind of site you are working with, see *Is my TeamForge site "dedicated" or "advanced?"* on page 59

1. Back up your site data.

   a) Make an archive file with the following data directories:

Directory	Contents
/svnroot	Subversion source code repositories
/cvsroot	CVS source code repositories (not present on all sites)

   ```
   mkdir /tmp/backup_dir
   cp -Rpf /svnroot /cvsroot /tmp/backup_dir
   ```

   b) Back up your SSH keys, if any.

   c) Back up your SSL certificates and keys, if any.

2. Stop TeamForge.

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

3. Move the repositories from any previous installs out of the way.

   ```
   mv /etc/yum.repos.d/collabnet-6.1.0.0.repo
   /etc/yum.repos.d/collabnet-6.1.0.0.repo.cn_backup
   ```

4. Download the TeamForge 6.1.1 installation repository from *open.collab.net*. Copy it to `/etc/yum.repos.d/`.

5. Uninstall the PostgreSQL support packages to clear the way for to install PostgreSQL. (These packages are installed with the OS by default.)

   ```
   yum erase postgresql-libs postgresql-docs
   ```

6. On the source code box, install the source code component of the TeamForge application.

   ```
   yum clean all
   yum install teamforge-scm
   ```

7. Make sure your Java settings are right for TeamForge 6.1.1.

   a) Check the *JAVA_HOME* variable in `/opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf` to make sure TeamForge is using the right JDK.

   ```
   vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
   ```

   The JAVA_HOME variable should look like this:

   ```
   JAVA_HOME=/usr/java/jdk1.6.0_26
   ```

   b) In the *JBOSS_JAVA_OPTS* variable, increase the MaxPermSize value to 512.

8. If you don't specify a domain name, replace the HOST_localhost token with the hostname in the `site-options.conf` file. Otherwise, ViewVC pages for Subversion and CVS repositories created after the site is up, will be rendered with a CSS error.

9. In the `site-options.conf` file, use the shared secret key, *SCM_DEFAULT_SHARED_SECRET*, from the `site-options.conf` file on the primary TeamForge server.

10. Create the runtime environment.

   ```
   cd /opt/collabnet/teamforge-installer/6.1.1.0
   ./install.sh -r -I -V
   ```

11. Update the file permissions on your site's data.

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

12. Swap in the new Apache configuration file.

```
cd /etc/httpd/conf
mv httpd.conf httpd.conf_old
cp httpd.conf.cn_new httpd.conf
/etc/init.d/httpd start
```

13. Start TeamForge.

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

Is my TeamForge site "dedicated" or "advanced?"

Check the value of the *DEDICATED_INSTALL* variable to see what kind of site you are working with.

The type of TeamForge installation you have makes a difference for how you upgrade and patch the site. If you weren't the one who installed your existing site, you'll need to find out if your site is a dedicated or advanced installation.

One easy way to tell if you have an advanced site is to check if any of your site's services are running on separate boxes from the main TeamForge application. This can only happen on an advanced site.

However, if your site has all its services running on the same box, it is not necessarily a dedicated site in the sense that we're talking about. You can have an advanced site on a single box.

If there is any doubt, look in the site's master configuration file.

1. Open the `site-options.conf` file.

This is the master configuration file that controls your TeamForge site.

```
vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
```

👉 **Note:** `vi` is an example. Any *nix text editor will work.

2. Look for the *DEDICATED_INSTALL* variable.

- If *DEDICATED_INSTALL* is set to true, the TeamForge you have a dedicated installation, with the default configuration and minimal user intervention.
- If *DEDICATED_INSTALL* is set to false, or is not present, you have an advanced installation, with customizations appropriate to this particular site's conditions and use patterns.

3. As you work through the instructions for upgrading or patching your site, watch for notes like this one:

👉 **Note:** If you are working with a *dedicated* TeamForge installation, you can skip this. See *Is my TeamForge site "dedicated" or "advanced?"* on page 59

If you do have a dedicated site, this may help lighten your load a little.

Enable reporting while upgrading to TeamForge 6.1.1

When upgrading from a TeamForge 6.1 site that does not have reporting, you have to configure a collection of variables in the `site-options.conf` to turn on reporting in TeamForge 6.1.1.

The following procedure is an outline of what you need to when upgrading from TeamForge 6.1 (no reporting) to TeamForge 6.1.1 (with reporting). For specific commands on upgrading (on the same or new hardware), see the instructions for *upgrade (dedicated)* and *upgrade (advanced)*.

1. Stop TeamForge.
2. Back up your site data.
3. Install the TeamForge 6.1.1 build, making sure that you change the 6.1 site-options for the `jdk` and `MaxPermSize` requirements in 6.1.1.

    ☞ **Note:** Tokens related to reporting should not be enabled.

4. Migrate the data to TeamForge 6.1.1.
5. Stop all services. *Enable reporting-related tokens in `site-options.conf`* and recreate the runtime.
6. Start the database if it is not running.
7. From the `runtime/scripts` directory, run the `bootstrap-reporting-data.py` script.
8. Start the TeamForge 6.1.1 site and make sure that reporting is enabled.

Troubleshooting: Upgrade PostgreSQL manually

The TeamForge upgrade utility upgrades PostgreSQL for you automatically when you run the `prepare-environment.sh` script. However, when you upgrade to TeamForge 6.1 from TeamForge 5.4, patch 1, it is possible for the automatic PostreSQL upgrade to fail. If this happens, you can do the PostgreSQL upgrade yourself.

The error, if it occurs, will look like this:

```
ERROR: postgres upgrade failed.
Exiting due to fatal error.
```

1. Create a database dump.

    ☞ **Note:** If you have already made the dump, skip this step.

    ```
    su - postgres -c /usr/bin/pg_dumpall > /tmp/dumppath/
    ```

2. Upgrade PostgreSQL.
    a) Ensure that PostgreSQL is running.

       ```
       /etc/init.d/postgresql-9.0 start
       ```

    b) Upgrade the PostgreSQL packages.

       ```
       yum install postgresql postgresql-server postgresql-docs
       ```

    c) If you get this error:

       ```
       %postun(postgresql-server-8.3.8-1PGDG.rhel5.x86_64) scriptlet failed,
       exit status 1
       ```

       then run this command:

       ```
       yum erase postgresql-server-8.3.8-1PGDG.rhel5
       ```

3. Move the old `pgsql` directory out of the way.

```
mv /var/lib/pgsql/9.0/ /var/lib/pgsql/9.0_old
```

4. Rerun the `prepare-environment.sh` command. When it runs successfully, you can run `install.sh` and `migrate.py`.

> ☞ **Important:** After you run the installer and before you run the migration command, you must reload the PostgreSQL data dump:
>
> ```
> su - postgres -c /usr/bin/psql < /tmp/dumppath/<name>.dmp
> ```

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 of TeamForge 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 *Upgrade to TeamForge 6.1.1 on Red Hat*

1. Stop TeamForge.

```
/etc/init.d/httpd stop
/etc/init.d/postgresql-9.0 stop
/etc/init.d/collabnet stop
```

2. Install the TeamForge application.

```
yum install teamforge-app teamforge-scm teamforge-database teamforge-etl
```

> ☞ **Tip:** If the yum installer balks, you may have duplicate rpm packages.
>
> **1.** Get the `yum-utils` package, if it isn't already installed.
>
> ```
> yum install yum-utils
> ```
>
> **2.** Check for duplicate packages.
>
> ```
> package-cleanup --dupes
> ```
>
> **3.** Clean up the older packages, if any.
>
> ```
> package-cleanup --cleandupes
> ```
>
> **4.** Rerun the yum installer.
>
> ```
> yum install teamforge [options]
> ```

3. Run the installer.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
```

> ☞ **Note:** Make sure the token SCM_DEFAULT_SHARED_SECRET is present in the `site-options.conf`.

```
sudo ./install.sh -r -I -V
```

4. Start TeamForge.

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

Convert TeamForge repositories from CVS to Subversion

To convert TeamForge repositories from CVS to Subversion, use the `ctf-cvs2svn.py` script.

1. Make sure Subversion is enabled on the TeamForge site.

2. Do a pre-flight scan of the RCS files.

```
sudo RUNTIME_DIR/scripts/cvs2svn.py -p 1:2 --dumpfile /dev/null
CVS_REPOSITORY_BASE/<cvs_repository>
```

3. Convert the CVS data into a Subversion dump file.

```
sudo RUNTIME_DIR/scripts/cvs2svn.py --dumpfile=/path/to/final/dumpfile
CVS_REPOSITORY_BASE/<cvs_repository>
```

If you are converting large repositories, you may encounter a memory error. If that happens, pass the `--use-rcs` option to the this command as follows:

```
sudo RUNTIME_DIR/scripts/cvs2svn.py --dumpfile=/path/to/final/dumpfile
--use-rcs CVS_REPOSITORY_BASE/<cvs_repository>
```

> 👉 **Note:** Like `--use-rcs`, there are various other options you can pass to `cvs2svn.py` as appropriate for the CVS repository you're converting.

4. Load the dump file and convert the project.

```
sudo RUNTIME_DIR/scripts/ctf-cvs2svn.py
--cvs-integration-name=<cvs_integration_name>
--svn-integration-name=<svn_integration_name>
--cvs-repository-name=<cvs_repository_name> --dumpfile=/path/to/dumpfile
```

When there is already a Subversion repository with the same name as the CVS repository, provide a Subversion repository name explicitly using the `--svn-repository-name` option.

```
sudo RUNTIME_DIR/scripts/ctf-cvs2svn.py
--cvs-integration-name=<cvs_integration_name>
--svn-integration-name=<svn_integration_name>
--cvs-repository-name=<cvs_repository_name> --dumpfile=/path/to/dumpfile
--svn-repository-name=<svn_repository_name>
```

> 👉 **Note:**
>
> - If the CVS repository has spaces, you must enclose it within quotes while running the script.
> - Every time you convert a repository, you need to restart the TeamForge site after the conversion.
> - In the case of an advanced mode TeamForge installation where SCM resides on a separate box, you need to add the <scm_host_ip> entry in the `pg_hba.conf` file of the TeamForge database.

Patch CollabNet TeamForge 6.1.1 on Red Hat

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

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

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

By default, the installation source directory is `/opt/collabnet/teamforge-installer/6.1.1.0`. If necessary, download the installer from the CollabNet web site and unpack it in this directory.

Apply a patch

The patch installer is built into the existing TeamForge installation, and it updates itself automatically via `yum`when you run it.

1. Stop TeamForge.

   ```
   /etc/init.d/httpd stop
   /etc/init.d/postgresql-9.0 stop
   /etc/init.d/collabnet stop
   ```

2. Run the patch installer from the base installer directory.

   ```
   cd /opt/collabnet/teamforge-installer/6.1.1.0
   ./patch-install.sh -l 1 -n -V -d /opt/collabnet/teamforge
   ```

   The -l parameter specifies the number of the patch (also called the patch level) and the -d parameter specifies the directory where the site is running.

   ☞ **Note:** If you have redirected the logs during the patch install process (`./patch-install.sh -l 2 -d /opt/collabnet/teamforge &> /tmp/install.log`), you may see the command prompt at this point. This does not indicate that the installation has ended. To verify that the install process is still running, you can run `ps -ef | grep install`.

3. Start TeamForge.

   ```
   /etc/init.d/httpd start
   /etc/init.d/postgresql-9.0 start
   /etc/init.d/collabnet start
   ```

4. Verify that your patch change has worked.
   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.

5. Ensure that your users can access their source code.
   a) Click **Admin** in the CollabNet TeamForge navigation bar.
   b) On the site administration navigation bar, click **Integrations**.
   c) For each source control service you are supporting, verify that the right paths are specified.

      • **SOAP service host** should be `localhost` or the host name of the server on which you just installed TeamForge.
      • **Repository base URL** should be the URL for the top level of your source code server (which may be the same as your application server). For example, `http://<myscmbox>/svn/repos`

- **SCM Viewer URL** should be the URL for the ViewVC application on your source control server. For example, `http://<myscmbox>/integration/viewvc/viewvc.cgi`

d) Select all your source code integrations and click **Synchronize Permissions**.

This updates the permissions on your code repositories so that users can access them from the new site.

☞ **Note:** By default, the *DISABLE_CREATE_INTEGRATION_SERVERS* flag in the `site-options.conf` file is set to false, which allows users to create new external integrations. To suppress the ability to add integrations, change this setting to true and recreate the runtime environment before making the site available to users.

☞ **Note:** During a patch upgrade, the TeamForge installer creates a local repository where it keeps stuff it's going to need. When the upgrade is finished, this local repository is kept for backup purposes.

If for any reason you need to run the patch procedure again, you can reuse this local repository, but you must first rename it so the installer can find it, like this:

```
mv cn-collabnet-local.repo.backup cn-collabnet-local.repo
```

Remove a patch

To downgrade a patch that you have installed on your TeamForge site, run the patch installer from the directory corresponding to that patch.

1. Stop TeamForge.

```
/etc/init.d/httpd stop
/etc/init.d/postgresql-9.0 stop
/etc/init.d/collabnet stop
```

2. Change to the install directory for the patch you want to revert.
   For example, if you are reverting from patch 3:

```
cd /opt/collabnet/teamforge-installer/6.1.1.3
```

3. Run the patch installer with the appropriate level flag.

```
./patch-install.sh -d /opt/collabnet/teamforge -l <single-digit number of
patch you are downgrading to> -V
```

For example, to downgrade to the base installation with no patches:

```
./patch-install.sh -d /opt/collabnet/teamforge -l 0 -V
```

4. Copy the `site-options.conf` file from the backup directory back to the installation directory for your new patch level.

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

cp 6.1.1.0-backup/conf/site-options.conf 6.1.1.0/conf/
```

5. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0

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

6. Start TeamForge.

```
/etc/init.d/httpd start
/etc/init.d/postgresql-9.0 start
/etc/init.d/collabnet start
```

7. Verify that the patch was reverted.

    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.

Downgrade a patch using YUM

To downgrade a patch on your TeamForge site using the YUM installer, you need to specify the TeamForge packages for the services enabled on the box and the list of CN-* packages that changed in that patch.

For the services enabled on the box, these are the TeamForge rpms you need to specify in the patch downgrade command:

- app: teamforge-app, teamforge-etl, teamforge-scm
- etl: teamforge-etl, teamforge-scm
- scm: teamforge-scm
- database/datamart: teamforge-database

1. Stop TeamForge.

```
/etc/init.d/httpd stop
/etc/init.d/postgresql-9.0 stop
/etc/init.d/collabnet stop
```

2. Find the packages contained in the patch.

For example, let's say you ran this command:

```
cat /opt/collabnet/teamforge/dist/conf/patches
```

and got this result:

```
118.1: CN-runtime,CN-jakarta-log4j,
118.2: CN-quartz,CN-runtime,
```

3. Run the downgrade command.

To downgrade from patch level 1 to patch level 0 in the above example, here's the command:

```
yum downgrade teamforge teamforge-app teamforge-etl teamforge-scm
teamforge-database TeamForge-installer CN-jakarta-log4j CN-runtime
```

To downgrade from patch level 2 to patch level 1 in the example, here's the command:

```
yum downgrade teamforge teamforge-app teamforge-etl teamforge-scm
teamforge-database TeamForge-installer CN-runtime CN-quartz
```

☞   **Note:** During the downgrade, the `site-options.conf` file is automatically backed up into `/opt/collabnet/teamforge-installer//6.1.1.0/conf`.

4. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0

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

5. Start TeamForge.

```
/etc/init.d/httpd start
/etc/init.d/postgresql-9.0 start
/etc/init.d/collabnet start
```

6. Verify that the patch was reverted.
   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.

Maintain your CollabNet TeamForge 6.1.1 site on Red Hat

You've got CollabNet TeamForge installed, up to date, and operating on the appropriate scale. Now you're in day-to-day maintenance mode. While you're waiting around for something to go wrong, there's a lot you can do to support users and project managers, monitor the site's performance, and generally keep things running smoothly.

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 is running on, as specified in your order form.

👉 **Tip:** These steps are for installing your license key via the web interface. If you prefer, you can install it as a text file instead. See *Supply your CollabNet TeamForge license key as a text file* on page 67.

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:

   25sgeviliairirc144.16.16.25.:32025108D953D35F565FA86BDAC174D613X0244D0453493H05100A63487AD8DF1325366574B57696A3B7231394231610263132E

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

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

Supply your CollabNet TeamForge license key as a text file

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 is running on.

👉 **Important:** If you are upgrading from a site with a limited number of users to an enterprise-scale site, you must install your license key before starting CollabNet TeamForge . Otherwise, your site could be rendered inoperable.

1. Locate the confirmation email you received from your CollabNet representative when you purchased your contract.
2. Create a text file and copy-paste your license key from the confirmation email into it.

   For example, if your organization has 80 users who will use only the source code management features and 100 users who need the full range of application lifecycle management features, your license key string may look like this:

   ala100S1480sgeviliairirc144.16.16.25.:32025108D953D35F565FA86BDAC174D613X0244D0453493H05100A63487AD8DF1325366574B57696A3B7231394231610263132E

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

3. Save the text file as `/opt/collabnet/teamforge/var/etc/sflicense.txt`

   👉 **Tip:** Save your license key somewhere remote too, in case you need to reinstall CollabNet TeamForge and your `sflicense.txt` file is not accessible.

4. Make the license file usable by the application.

   ```
   chmod 0400 /opt/collabnet/teamforge/var/etc/sflicense.txt
   chown ctf-admin:ctf-admin /opt/collabnet/teamforge/var/etc/sflicense.txt
   ```

   👉 **Note:** If necessary, change `ctf-admin` to match the value of the *APP_USER* variable in the `site-options.conf` file.

Support CollabNet TeamForge users

As a system administrator, you can do these things to help maximize the effectiveness and productivity of your site's users.

Authenticate users with LDAP

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

Set up LDAP integration for the CollabNet TeamForge site

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

☞ **Note:** Only OpenLDAP 2.3.27-5 is officially supported, but with some modifications you may be able to make a simple Active Directory integration work as well. See the suggested modifications for *login-config.xml* on page 148.

1. Shut down CollabNet TeamForge .

   ```
   /etc/init.d/httpd stop
   /etc/init.d/collabnet stop
   /etc/init.d/postgresql-9.0 stop
   ```

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

   ```
   cd /opt/collabnet/teamforge
   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 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 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 `/opt/collabnet/teamforge/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, you will not be able to create the user in CollabNet TeamForge .

4. In the `/opt/collabnet/teamforge/var/etc/login-config.xml` file, modify the TeamForge `application-policy` block to enable CollabNet TeamForge to authenticate against your LDAP server.

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

   a) Replace the TeamForge `application-policy` block with the code listed in the sample `application-policy` block in *login-config.xml* on page 148.

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 /opt/collabnet/teamforge
```

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, you can configure the integration server to use OpenLDAP 2.3.27-5.

👉 **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.

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

Option	Value
suffix	"dc=teamforge,dc=com"
rootdn	"cn=Admin,dc=teamforge,dc=com"
rootpw	sfee
sizelimit	30000

👉 **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} 7hC2H5OoEZOaT6rL3hAvyxy11jrZYB2
```

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

8. Add the LDIF information.

```
ldapadd -x -D cn=Admin,dc=teamforge,dc=com -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 an old TeamForge version 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.
    a) Open the authentication GUI tool with the `authconfig` command.
    b) Select **Use LDAP**.
    c) Specify the base DN and server.

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

    d) Click **Next**.
    e) Select **Use LDAP Authentication**. Fill in the DlDAP basedn and host information.
    f) 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 *passws*, *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=teamforge,dc=com?one nss_base_shadow
ou=Users,dc=teamforge,dc=com?one nss_base_group
ou=Groups,dc=teamforge,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=teamforge,dc=com	Schema address that users are added into.
CVS_LDAP_GROUPS_DN	Example: ou=Groups,dc=teamforge,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=teamforge,dc=com	The "root dn" for the LDAP server. This must be the value specified in `slapd.conf`.
CVS_LDAP_BIND_PASSWORD		The password for the root dn.
INTEGRATION_OS	linux_ldap	
CVS_USER_DEFAULT_GROUP	sfee	

16. Restart the integration server.

```
/opt/collabnet/teamforge/runtime/scripts/teamforge-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 SoureceForge 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 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 , 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 configuration and enable your CollabNet TeamForge JBoss installation to authenticate against your corporate LDAP server, you must restart CollabNet TeamForge .

Turn off LDAP authentication

During some maintenance operations, such as upgrades, you may need to turn off LDAP authentication temporarily.

1. Open the `site-options.conf` file.

    This is the master configuration file that controls your TeamForge site.

    ```
    vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
    ```

    👉 **Note:** `vi` is an example. Any *nix text editor will work.

2. In the `site-options.conf` file, comment out these variables:

    *   USE_EXTERNAL_USER_AUTHENTICATION
    *   LOGIN_CONFIG_XML_FILE

- MINIMUM_PASSWORD_LENGTH

3. Restart the runtime environment.

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

4. Review the variables you've changed, then save the `site-options.conf` file.

Back up CollabNet TeamForge data
Save a copy of your TeamForge site's data to a location where you can quickly retrieve it to your TeamForge 6.1.1 site.

- If you are upgrading by installing TeamForge 6.1.1 on new hardware, then you'll need the backed-up site data to complete the upgrade.
- If you are upgrading your site on the same hardware, then you won't need the backed-up data, but you should create it anyway, as a precaution.

1. Stop the TeamForge application server and the Apache server, if they are running.

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

2. Back up your site data.

a) Make a dump file of your site database.

(This may be the same as your TeamForge application server or a separate box.)

You have to do a PostgreSQL dump because we are upgrading the PostgreSQL application as part of this upgrade.

☞ **Note:** These commands are for a PostgreSQL database, which is the default. If your site uses an Oracle database, follow the *Oracle backup procedure* instead.

```
su - postgres
/usr/bin/pg_dumpall > /var/lib/pgsql/backups/teamforge_data_backup.dmp
exit
mkdir /tmp/backup_dir
cp /var/lib/pgsql/backups/teamforge_data_backup.dmp /tmp/backup_dir/
```

☞ **Note:** If your reporting database is running on a separate port, back up your reporting database too:

```
/usr/bin/pg_dumpall -p <reports_database_port> >
/var/lib/pgsql/backups/teamforge_reporting_data_backup.dmp
```

b) Make an archive file with the following data directories:

Directory	Contents
/opt/collabnet/teamforge/var	User-created data, such as artifact attachments
/svnroot	Subversion source code repositories
/sf-svnroot	Subversion repository for branding data
/cvsroot	CVS source code repositories (not present on all sites)

```
cp -Rpf /svnroot /sf-svnroot /cvsroot /opt/collabnet/teamforge/var
/tmp/backup_dir
cd /tmp
tar czvf 61backup.tgz backup_dir
```

     c) Back up your SSH keys, if any.

     d) Back up your SSL certificates and keys, if any.

Set up an Oracle database

To use an Oracle database for your CollabNet TeamForge 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:** Replace <sf user> with the database username specified in `site-options.conf` and <sf passwd> with the database password specified in `site-options.conf`.

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

Rebuild TeamForge search indexes

You can rebuild your site's search index without stopping TeamForge.

Any new objects created during this time will not be immediately indexed, but will be queued until after the re-indexing.

1. Make sure the TeamForge site is up.

2. Run the re-index script.

   ```
   /opt/collabnet/teamforge/runtime/scripts/SearchReindex.py
   ```

After the script completes, everything is queued for re-indexing. It will take some time to process the re-index requests.

👉 **Note:** Due to the Lucene upgrade for search functionality, upgrading to TeamForge 6.1.1 requires a complete re-index of the site. This could take several hours, and the index data could double in size.

Permit big file uploads

When many users store very large files on your site, you may sometimes notice a slowdown in your site's performance. You can reduce the impact of such a use pattern by telling TeamForge not to index files larger than a certain size.

👉 **Note:** It's also a good idea to let your users know that the Documents tool in TeamForge is not designed primarily as a storage device. As a best practice, upload documents to make them available for collaboration, not for backup or long-term storage.

1. Open the `site-options.conf` file.

   This is the master configuration file that controls your TeamForge site.

   ```
   vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
   ```

   👉 **Note:** `vi` is an example. Any *nix text editor will work.

2. Add the *SEARCH_MAX_FILE_SIZE* parameter and give it a value equal to the maximum size (in bytes) of files to be indexed.

   👉 **Note:** The default value is 10M. With this value, files larger than 10M are not indexed.

   A value of zero or less specifies that there is no limit, which is the same as the default behavior without the variable.
3. Review the variables you've changed, then save the `site-options.conf` file.

Allow searching of archive files

By default, users can't search the content of archive files uploaded to TeamForge, such as `zip`, `tar`, or `docx` files. If your users need it, you can provide this ability.

Microsoft Office 2007 files, such as files with the `.docx` extension, are archive files. By default their content is not indexed and does not show up in search results. However, information that TeamForge maintains about those documents, such as title, author, description and version, does appear in search results.

If you permit archive searching, watch for performance slowdowns associated with the larger volume of indexing that TeamForge is doing. Depending on your site members' use patterns, the performance cost may or may not be acceptable to your users.

👉 **Note:** It's also a good idea to let your users know that the Documents tool in TeamForge is not designed primarily as a storage device. As a best practice, upload documents to make them available for collaboration, not for backup or long-term storage.

1. Open the `site-options.conf` file.

   This is the master configuration file that controls your TeamForge site.

   ```
   vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
   ```

   👉 **Note:** `vi` is an example. Any *nix text editor will work.

2. Add the *SEARCH_SUPPRESS_ARCHIVE_SUB_DOCS* parameter and set it to false.
   This enables files inside archives (such as `.zip`, `.gz`, or `.tar`) to be indexed for search.
3. Review the variables you've changed, then save the `site-options.conf` file.

Limit the size of message attachments

To avoid overtaxing your mail server or your storage volume, you may want to set a ceiling on the size of the attachments that users can send to a forum via email.

When a user sends an attachment that is larger than the limit, the message is rejected and the user gets an email from the Site Administrator explaining that the attachment exceeded the limit.

> ☞ **Tip:** Before imposing a file attachment size limit, it's a good idea to point your users to better ways of collaborating around large files. Consider suggesting source code repositories, backup systems, or other appropriate solutions.

1. Open the `site-options.conf` file.

   This is the master configuration file that controls your TeamForge site.

   ```
   vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
   ```

   > ☞ **Note:** `vi` is an example. Any *nix text editor will work.

2. Set the value of the *DISCUSSION_MAX_ATTACHMENT_SIZE* variable to a reasonable level.
   For example, if your users are given to using Microsoft Word documents on the site, you might set *DISCUSSION_MAX_ATTACHMENT_SIZE* to 10 MB, and increase the value by two or three MB at a time if users need more headroom.
3. Review the variables you've changed, then save the `site-options.conf` file.

Limit the size of document attachments

When many users store very large documents on your site, you may sometimes notice a slowdown in your site's performance. You can reduce the impact of such a use pattern by telling TeamForge not to attach documents larger than a certain size.

> ☞ **Note:** It's also a good idea to let your users know that the Documents tool in TeamForge is not designed primarily as a storage device. As a best practice, upload documents to make them available for collaboration, not for backup or long-term storage.

1. Open the `site-options.conf` file.

   This is the master configuration file that controls your TeamForge site.

   ```
   vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
   ```

   > ☞ **Note:** `vi` is an example. Any *nix text editor will work.

2. Add the *DOCUMENT_MAX_FILE_UPLOAD_SIZE* parameter and give it a value equal to the maximum size (in megabytes) of documents to be uploaded.
3. Review the variables you've changed, then save the `site-options.conf` file.

Who can post to discussions by email?

To help reduce the risk of spam or other mischief, you may need to limit the users who can post to discussion forums by email.

To leverage the advantages of community collaboration, you should keep your forums as open as you can. However, some sites require tighter control over who can participate in discussions. TeamForge enables you to balance openness against privacy along a spectrum of choices.

1. Open the `site-options.conf` file.

   This is the master configuration file that controls your TeamForge site.

```
vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
```

👉 **Note:** `vi` is an example. Any *nix text editor will work.

2. Add the *DISCUSSION_EMAIL_POSTING* variable, and give it the value that reflects the degree of openness your site's discussion forums require.

Use one of these values:

Value	Description
0	Allow only forum admins.
1	Users with roles and permissions.
4	All logged in users.
5	Allow known email addresses only.
6	Allow all site users and guests.

3. Review the variables you've changed, then save the `site-options.conf` file.

The value you set here determines the maximum degree of openness to email posting for all projects on your site. For example, consider a site where project members can post by email (level 3). For a project that requires extra security, the project administrator can choose to accept email only from users with the appropriate role (level 1). However, a project owner cannot accept email posts from a less restrictive category of users, such as all users who are logged in (level 4).

Who can monitor discussions?
You may need to limit the users who can monitor discussion forums.

To leverage the advantages of community collaboration, you should keep your forums as open as you can. However, some sites require tighter control over how users keep track of discussions. TeamForge enables you to balance openness against privacy along a spectrum of choices.

1. Open the `site-options.conf` file.

This is the master configuration file that controls your TeamForge site.

```
vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
```

👉 **Note:** `vi` is an example. Any *nix text editor will work.

2. Add the *DISCUSSION_EMAIL_MONITORING* variable, and give it the value that reflects who you want to get updates on discussions via email.

Use one of these values:

Value	Description
0	Allow only forum admins.
1	Users with role permissions.
4	All logged in users.
5	Allow all site users and guests.

3. Review the variables you've changed, then save the `site-options.conf` file.

The value you set here determines the maximum degree of openness to monitoring discussions for all projects on your site. For example, consider a site where project members can monitor discussions (level 3). For a project that requires

extra security, the project administrator can dictate that only users with the appropriate role can monitor discussions (level 1). However, a project owner cannot allow monitoring for a less restrictive category of users, such as all users who are logged in (level 4).

Reduce discussion spam

You can filter out some kinds of spam from your project's discussion forums.

1. Open the `site-options.conf` file.

   This is the master configuration file that controls your TeamForge site.

   ```
   vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
   ```

   👉 **Note:** `vi` is an example. Any *nix text editor will work.

2. Add one or more MIME types to the Reject MIME types filter.

   The presence of any of these MIME types in an incoming message (via email) will cause its rejection with appropriate notification to the posting user.

   For example:*DISCUSSION_REJECT_MIME_TYPES*=application/pdf,text/xml

3. Add one or more MIME types to the Drop MIME types filter.

   The presence of any of these MIME types in an incoming message (via email)will cause its attachment to be deleted with appropriate notification to the posting user.

   For example:*DISCUSSION_DROP_MIME_TYPES*=image/jpeg,image/jpg,text/xml

4. Add one or more header names to the remove headers filter.

   If an incoming email posting contains any of these headers, they will be quietly removed from the message before it is archived and before subscribers are notified.

   For example:*DISCUSSION_REMOVE_HEADERS*=precedence,x-no-archive,Return-Path

5. Add one or more header names to reject headers filter to be rejected or moderated (if discussion is moderated).

   Use regular expressions, each regular expression must match an entire header. The match of any of these headers in an incoming message (via email) will cause its rejection with appropriate notification to the posting user.

   For example:*DISCUSSION_REJECT_HEADERS*=(?s).*headername1:value2.*,(?s).*name2:value2.*

6. Add one or more entries for Reject content filter.

   Use regular expressions, each regular expression must match an entire entry. The match of any of these entries in discussion body and subject of an incoming message (via email) will cause its rejection with appropriate notification to the posting user.

   For example: *DISCUSSION_REJECT_CONTENT*=(?s).*word.*,(?s).*spam.*

   👉 **Note:** The content entry is a case sensitive.

7. Review the variables you've changed, then save the `site-options.conf` file.

8. Restart the site.

Install project templates

Provide sample projects to help users get started quickly.

TeamForge comes with a sample template useful for agile development projects. Site administrators and project managers can use this template to jump-start a project without a lot of manual setup steps.

In the TeamForge installation directory, run the `install-project-templates.py` script.

```
cd /opt/collabnet/teamforge-installer/6.1.1.0
./install-project-templates.py -V
```

Use a site administrator user name and password. For a new site, these are `admin` and `admin`.

👉 **Tip:** On some servers, it may take a few seconds for the SOAP server to be ready after installation. If `install-project-templates.py` returns an error, try waiting briefly and then running it again.

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.

By default, only site administrators can see project template detail, but project administrators normally create a project from a project template. To choose the right template, a project manager may want to know if tasks, documents, wiki pages or other kinds of content are included in a given template.

👉 **Note:**

1. Open the `site-options.conf` file.

   This is the master configuration file that controls your TeamForge site.

   ```
   vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
   ```

   👉 **Note:** `vi` is an example. Any *nix text editor will work.

2. Change the value of the SHOW_PROJECT_TEMPALTE_DETAIL_TO_NON_SITEADMINS variable to true.

   👉 **Note:** Use the parameter name as given, including the typo.

3. Review the variables you've changed, then save the `site-options.conf` file.
4. Recreate the runtime environment.

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

Provide a Perforce source control server

Enabling your users to integrate Perforce repositories into their TeamForge projects requires some extra configuration.

1. If you are adding Perforce support to an existing site, back up your site's data first. (If you are adding Perforce as part of installing a new site, skip this.)

   See *Back up CollabNet TeamForge data* on page 94.

2. Install Perforce, using the instructions in the *Perforce sysadmin documentation*.
3. Open the `site-options.conf` file.

   This is the master configuration file that controls your TeamForge site.

   ```
   vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
   ```

   👉 **Note:** `vi` is an example. Any *nix text editor will work.

4. Add these variables to your `site-options.conf` file, changing the values as appropriate:

   ```
   PERFORCE_CLIENT=/usr/local/bin/p4
   PERFORCE_PORT=localhost:1666
   PERFORCE_LICENSE_FILE=/tmp/license
   ```

5. Review the variables you've changed, then save the `site-options.conf` file.
6. If your Perforce server is running SuSE, remove the `perforce` user from the TeamForge server and bootstrap the site data. (If you are on Red Hat or CentOS, skip this.)

   ```
   userdel perforce
   ```

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

7. If you are adding Perforce support to an existing site, restore your site's data.

See *Restore backed-up CollabNet TeamForge data* on page 94.

Protect your CollabNet TeamForge site

You can take various measures to maximize the security of your CollabNet TeamForge users.

Set up SELinux

If SELinux is active on the machine where your CollabNet TeamForge site is running, modify it to allow the services that TeamForge requires.

This type of configuration is required for a complete TeamForge installation.

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

```
setsebool -P httpd_can_network_connect 1
```

2. When you are installing TeamForge, follow these steps to enable services to come up when needed.

a) Change the context for the TeamForge log.

```
chcon -R -h -t httpd_sys_content_t /opt/collabnet/teamforge/log/httpd
```

b) Change the context for PostgreSQL.

```
chcon -R -h -t postgresql_db_t /var/lib/pgsql/9.0
```

c) Create the PostgreSQL log directory and set the appropriate permissions.

```
mkdir /opt/collabnet/teamforge/log/pgsql
chown -R postgres:postgres /opt/collabnet/teamforge/log/pgsql
chmod 777 /opt/collabnet/teamforge/log/pgsql/
```

d) Change the context for the PostgreSQL log.

```
chcon -R -h -t postgresql_db_t /opt/collabnet/teamforge/log/pgsql/
```

3. After your site is installed and contains some live data, follow these steps to allow source control services to work.

a) Change the context for your Subversion source code service.

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

b) 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 /opt/collabnet/teamforge/var/overrides
```

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

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

Protect your TeamForge site with SSL
Use Secure Socket Layer (SSL) to run your Web server securely.
Set up SSL for your TeamForge site
To force all TeamForge 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. Open the `site-options.conf` file.

   This is the master configuration file that controls your TeamForge site.

   ```
   vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
   ```

   ☞　　**Note:** `vi` is an example. Any *nix text editor will work.

3. Set the options to enable SSL for the site.
   a) Set the *SSL* variable to on.
   b) Set 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 *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.

4. In the `site-options.conf` file, make sure the value of the *DOMAIN_localhost* variable matches that of your SSL certificate.

5. Rename the `/etc/httpd/conf.d/ssl.conf` file to `/etc/httpd/conf.d/ssl.conf.old`, if it exists.

6. If you are converting an existing site to use SSL (that is, if your site already has had users accessing it via HTTP and not HTTPS), you must update your site's publishing repository to use the new SSL settings.

   To do this, ask your CollabNet support representative for the `fix-publishing-repos-to-ssl.py` script.

7. Stop TeamForge.

   ```
   /etc/init.d/httpd stop
   /etc/init.d/postgresql-9.0 stop
   /etc/init.d/collabnet stop
   ```

8. Run the installer.

   ```
   cd /opt/collabnet/teamforge-installer/6.1.1.0
   ./install.sh -r -I -V
   ```

9. Rename the `/etc/httpd/conf/httpd.conf.cn_new` file to `httpd.conf`, if it exists.

10. Start TeamForge.

   ```
   /etc/init.d/httpd start
   /etc/init.d/postgresql-9.0 start
   /etc/init.d/collabnet start
   ```

A new Apache configuration file is created with the information you provided in the `site-options.conf` file. The new file is named `httpd.conf.cn_new`. It contains `VirtualHost` sections for port 80 and port 443. All port 80 requests are redirected to port 443.

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

Generate 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 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.

2. In each section, add the following two lines:

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

3. 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 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.

1. 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.

2. 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`.

3. Locate the Java `keytool` utility.

This is `PATH_TO_JAVA/bin/keytool`

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

4. 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>
```

👉 **Note:** Any value is accepted for server in -alias <server>.

5. At the password prompt, use `changeit`.

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

6. 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.

7. Update /etc/sourceforge.properties to enable secure communication.

a) Set *sfmain.integration.listener_ssl* to true.

b) Set *sfmain.integration.listener_port* to 443.

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

9. Restart TeamForge

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

Set up external applications for a TeamForge site

When you integrate an external application into your TeamForge site, the site administrator can make that application available to users as if it were a native TeamForge component.

Install Pebble

To see what you can do with an integrated application in TeamForge, start by installing Pebble, a blogging application that you can configure to work as part of your TeamForge site.

1. Get the Pebble installer package from *open.collab.net* and unzip it.
2. Modify these values in the *installer/install.conf* file to suit your installation environment.

Options	Description
pebble.base.dir	Path where you want Pebble to be installed on this host.
ctf.baseurl	Absolute URL of the TeamForge site that you want to associate to, such as `https://my.ctf.instance/`
tomcat.port	Pebble runs on Tomcat. This token indicates which port you want Tomcat to be running on. Make sure there are no other services running on that port.
domain	The base url to be used for Pebble, such as `my.ctf.instance`. (You don't need `http:` or `https:` here.)
timezone	The time zone Pebble will use to timestamp blog entries.
java_home	Path to a JDK 1.6.x instance.
protocol	`http` if SSL is not being used; `https` if SSL is being used.
data.dir	Path in the file system where Pebble blogs will be stored.

3. Run the installer.

```
sudo python install.py -i -r
```

4. Set up the initial blog data.

```
sudo python bootstrap-data.py
```

This is known as "bootstrapping" the application.

> 👉 **Tip:** You can bootstrap again if you want to start from scratch, but any existing blogs will be deleted if you do.

5. Restart the Pebble application.

```
/etc/init.d/pebbled stop
/etc/init.d/pebbled start
```

You should now have a working Pebble instance ready to work with TeamForge. The installer has created two configuration files: `installer/conf/pebble-app.xml` and `installer/conf/pebble-dep.xml`. See *Integrate Pebble into your TeamForge site* on page 85 for how to use them.

Integrate Pebble into your TeamForge site

When the sample Pebble blogging application has been installed on your site, you can make it available for projects on your TeamForge site.

Pebble must be installed and configured before you can integrate it into your TeamForge site. See *Install Pebble* on page 84.

When you have integrated Pebble, projects on your site can add Pebble to their set of collaboration tools. The blogs they create will share many of the core TeamForge features, such as authorization, authentication, go-urls, association, linkification, templating, Project Pages components, and source code management support.

1. Log into TeamForge as an admin user.

2. Click **Integrated Apps** in the Site Administration toolbar.

3. Click **Create**.

4. Use the **Browse** window to find the two configuration files that enable the Pebble application to work as a part of TeamForge:

   - `pebble-app.xml` (Application configuration file): Contains the text strings for the Pebble user interface.
   - `pebble-dep.xml` (Deployment configuration file): Contains the data that Pebble needs to interact with TeamForge.

   Click **Next**.

5. On the **Preview** screen, review the parameters you set in the configuration files.

   ☞ **Note:** You may have to revise one or more values to ensure they are valid.

6. Click **Save**.

The Pebble application is now available for all projects on your site. You can direct project administrators to *the project admin help* for instructions on adding it to their own project toolbars.

☞ **Note:** You may need to adjust your site's look and feel to support your integrated application. See *the site admin help* for details.

Get information about a CollabNet TeamForge site

Use the `snapshot.py` utility to determine what processes are running on your CollabNet TeamForge 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.

   ```
   /opt/collabnet/teamforge/runtime/scripts/snapshot.py
   ```

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

- JBoss
- Tomcat
- James
- PostgreSQL
- Apache
- C6Migration

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.

Rebuild runtime without the install directory

You should keep your teamforge-installer directory around after installing TeamForge in case you need it later. However, if you delete or lose the directory you can still rebuild the application runtime.

1. Make a copy of the runtime-options.conf to use as the new site-options.conf file.

   ```
   cp /opt/collabnet/teamforge/runtime/conf/runtime-options.conf
   /var/tmp/site-options.conf
   ```

2. Rebuild the application runtime using the create-runtime.py script.

   ```
   /opt/collabnet/teamforge/dist/scripts/create-runtime.py -d
   /opt/collabnet/teamforge/ -f /var/tmp/site-options.conf
   ```

Turn on site-wide reporting

To use the site-wide reporting functionality, you have to configure a collection of variables in the site-options.conf file.

👉 **Tip:** For a view of what this looks like in action, see any of the advanced installation scenarios under *Install TeamForge the advanced way* on page 11.

👉 **Note:** Site administrators can view the status of the ETL server in **System Tools** > **Server Status**. The status displays the following values:

   • "OK" if enabled and running
   • "N/A" when disabled
   • "Could not connect" when enabled and not running

1. Open the site-options.conf file.

   This is the master configuration file that controls your TeamForge site.

   ```
   vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
   ```

   👉 **Note:** vi is an example. Any *nix text editor will work.

2. Comment out the default *HOST_localhost=app database subversion cvs* variable, then add (or uncomment) the variable that has the reporting services enabled:

   ```
   HOST_localhost=app database subversion cvs etl datamart
   ```

   👉 **Note:** The app attribute is the only one that must be present for this variable. The other attributes may be assigned to other HOST variables, if those services are running on different machines. See *Install TeamForge the advanced way* on page 11 for a detailed view of how this works.

3. Add these variables (or uncomment them, if they are already present) and give them the appropriate values:

   ```
   REPORTS_DATABASE_NAME=ctfrptdb
   REPORTS_DATABASE_PASSWORD=ctfrptpwd
   REPORTS_DATABASE_USERNAME=ctfrptuser
   REPORTS_DATABASE_READ_ONLY_USER=ctfrptreadonly
   REPORTS_DATABASE_READ_ONLY_PASSWORD=rptropwd
   ```

4. If you want your datamart to use a different port than your site database, you can also add the *REPORTS_DATABASE_PORT* variable:

```
REPORTS_DATABASE_PORT=5632
```

5. To enable Web services access to your datamart, add these variables:

```
ETL_SOAP_SHARED_SECRET=<arbitrary_string>
SOAP_ANONYMOUS_SHARED_SECRET=<arbitrary_string>
```

☞ **Tip:** For more information about configuring variables, see *site-options.conf* on page 112.

6. Set the time you want the reporting data to be collected, if different from the default 2:30 a.m. (local time).

```
ETL_JOB_TRIGGER_TIME=<cron expression>
```

☞ **Tip:** This value is a cron expression, not a time value. See *ETL_JOB_TRIGGER_TIME* on page 122 for more.

7. Review the variables you've changed, then save the `site-options.conf` file.

Synchronize TeamForge source control integrations

Any time you upgrade your TeamForge site or a source control application, you must ensure that your users can still access their source code.

1. Click **Admin** in the CollabNet TeamForge navigation bar.

2. On the site administration navigation bar, click **Integrations**.

3. For each source control service you are supporting, verify that the right paths are specified.

- **SOAP service host** should be `localhost` or the host name of the server on which you just installed TeamForge.
- **Repository base URL** should be the URL for the top level of your source code server (which may be the same as your application server). For example, `http://<myscmbox>/svn/repos`
- **SCM Viewer URL** should be the URL for the ViewVC application on your source control server. For example, `http://<myscmbox>/integration/viewvc/viewvc.cgi`

4. Select all your source code integrations and click **Synchronize Permissions**.

This updates the permissions on your code repositories so that users can access them from the new site.

☞ **Note:** By default, the *DISABLE_CREATE_INTEGRATION_SERVERS* flag in the `site-options.conf` file is set to false, which allows users to create new external integrations. To suppress the ability to add integrations, change this setting to true and recreate the runtime environment before making the site available to users.

Provide more than one source control server

To run more than one source control server of the same type on your site, each integration must have its own unique name and data directory.

☞ **Note:** No single server can host more than one source control integration of the same type. If you want to have more than one Subversion integration, they must run on separate machines. The same is true for CVS integrations.

1. Manually create the necessary directories on NetApp.

☞ **Important:** Each directory must have its own unique name. For example, if the first Subversion instance is named `svnroot`, you might name the second instance `svnroot_2`.

2. Set the permissions on the new directories to `ctf-admin:ctf-admin`

3. For each such directory on the NetApp, create a separate symlink in the local filesystem pointing to the NetApp mount folder.
   For example, assuming the NetApp mount is mounted on the `/shared` mount point in the local filesystem:

   ```
   sudo ln -s /svnroot /shared/svnroot_2
   ```

   👉 **Note:** Only one source code integration of any one type can run on a machine.

Upgrade Subversion

Use the yum package manager to upgrade to the latest supported Subversion release.

TeamForge 6.1.1 supports Subversion 1.7.2.

Details of the Subversion repository file:

```
FileName: subversion-1.7.2.repo
Content:
[CollabNetSVN]
name=collabnet_svn
baseurl=http://packages.collab.net/6.1.2.0/redhat/5Server/$basearch
gpgkey=http://packages.collab.net/RPM-GPG-KEY-collabnet
enabled=1
gpgcheck=0
```

1. Login as `root` and stop all `collabnet` services.

2. Copy file `subversion-1.7.2.repo` to `/etc/yum/repos.d/`

3. Check that Subversion 1.7.2 is available for upgrade:

   ```
   yum list subversion
   ```

4. Upgrade Subversion.

   ```
   yum update subversion subversion-perl subversion-python mod_dav_svn
   ```

5. Verify the Subversion upgrade.

   ```
   rpm -qa | grep subversion
   svn --version | grep " version"
   ```

6. Change to the `runtime/scripts` directory.

   ```
   cd /opt/collabnet/teamforge/runtime/scripts
   ```

   👉 **Note:** In the case of a multi-box setup, where SCM resides on a separate server, skip the next step and go directly to the last step to restart `collabnet` services.

7. Start all `collabnet` services.

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

Upgrade Subversion in CEE to CTF 6.1.1 migrated sites

TeamForge 6.1.1 supports Subversion 1.7.2

These are the steps to upgrade Subversion in CEE to TeamForge 6.1.1 migrated sites.

1. Stop `collabnet` services.

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

2. Uninstall the bdb instance set.

   ```
   cd /var/ops/teamforge-installer/6.1.1.0
   sudo ./instance-set-install.sh -d /opt/collabnet/teamforge -f
   bdb.svn.httpd-inst-x86_64-manifest-6.1.1.0.175-1 -u
   ```

   👉 **Note:** Instance set uninstallation successful. Do you want to start collabnet services? [YES/NO] NO

Details of the Subversion repository file:

```
FileName:subversion-1.7.2.repo
Content:
[CollabNetSVN]
name=collabnet_svn
baseurl=http://packages.collab.net/6.1.2.0/redhat/5Server/$basearch
gpgkey=http://packages.collab.net/RPM-GPG-KEY-collabnet
enabled=1
gpgcheck=0
```

3. Copy file `subversion-1.7.2.repo` to `/etc/yum.repos.d/`
4. Install the bdb instance set.

   ```
   cd /var/ops/teamforge-installer/6.1.1.0
   sudo ./instance-set-install.sh -d /opt/collabnet/teamforge -f
   bdb.svn.httpd-inst-x86_64-manifest-6.1.1.0.175-1
   ```

   👉 **Note:** Instance set uninstallation successful. Do you want to start collabnet services? [YES/NO] NO

5. Verify the Subversion upgrade.

   ```
   rpm -qa | grep CTF-subversion
   svn --version | grep " version"
   ```

   👉 **Important:** The next 2 steps are required only if Subversion is upgraded on the app server. In case of a multi-box setup, where SCM resides on a separate server, skip the next 2 steps and go to the last step to restart collabnet services.

6. Run the Subversion upgrade script.

   ```
   cd /opt/collabnet/teamforge/runtime/scripts
   sudo ./svn-upgrade-working-copies.sh
   ```

   👉 **Note:** Do the following verifications to upgrade to Subversion 1.7.2

Change the directory to

```
cd /opt/collabnet/teamforge/var/publish/<Any project's publishing
directory>/www
```

```
sudo /usr/sbin/svn --version
sudo /usr/sbin/svn --info
```

7. Set the permissions for branding and publish directories.

```
cd /opt/collabnet/teamforge/runtime/scripts
sudo eval `./options-to-shell-variables.py
sudo chown $HTTPD_USER:$HTTPD_GROUP $OVERRIDES_DIR $PUBLISH_DIR -R
```

8. Start all collabnetservices.

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

Change your site's domain name

To change the domain name of your site, you must change the name in the file system, the database, and any integrated applications. TeamForge provides a script for each of these.

1. Make sure your site's database and file system are backed up.
2. In the site-options.conf file, update the domain name (and hostname if needed), then save the file.

```
DOMAIN_<Host_Name>=<Domain_Name>
JAMES_POSTMASTER_EMAIL=root@<Domain_Name>
JAMES_MTA_HOST=<Domain_Name>
```

For example:

```
DOMAIN_mybox.supervillain.net=www.smileyface.com
JAMES_POSTMASTER_EMAIL=root@www.smileyface.com
JAMES_MTA_HOST=www.smileyface.com
```

3. Go to the TeamForge scripts directory.

```
cd /opt/collabnet/teamforge/runtime/scripts/
```

4. Run the script to change the domain in the file system.

```
./domain_change_fs.pl --old=www.myoldsitename.net --new=www.mynewsitename.net
> /tmp/domain_change_fs.out 2>&1
```

5. Run the script to change the domain in the database.

```
./domain_change_db.pl --old=www.myoldsitename.net --new=www.mynewsitename.net
> /tmp/domain_change_db.out 2>&1
```

6. If your site has Project Tracker integrated, run the script to change the domain in Project Tracker.

```
./domain_change_pt.py --oldDomain=www.myoldsitename.net
--newDomain=www.mynewsitename.net > /tmp/domain_change_pt.out 2> &1
```

Specify DNS servers

Define the DNS servers with which you want CollabNet TeamForge 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 /opt/collabnet/teamforge
   ```

3. Restart CollabNet TeamForge .

   ```
   /etc/init.d/httpd start
   /etc/init.d/postgresql-9.0 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 or at minimal server load times.

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

   ```
   vacuumdb -h `hostname` -U <DATABASE_USERNAME> -z <DATABASE_NAME>
   ```

   For example, using the default values in the `site-options.conf` file:

   ```
   vacuumdb -h `hostname` -U ctfuser -z ctfdb
   ```

2. 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.

1. Stop the site.

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

2. In the `/opt/collabnet/teamforge-installer/6.1.1.0/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.
3. Recreate the runtime environment.

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

4. Start the site.

   ```
   /etc/init.d/httpd start
   /etc/init.d/postgresql-9.0 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.

Change the logging level on your site

To support troubleshooting, you may need to set the logging level appropriately. Here's how to do that.

1. Edit `$RUNTIME/jboss/server/default/conf/log4j.xml`
2. Locate the `VAFILE` appender that starts with this line:
```
<appender name="VAFILE" ...
```

3. Change the value of the LevelMin parameter to INFO, DEBUG. ERROR, or WARN.
   For example:
```
<filter class="org.apache.log4j.varia.LevelRangeFilter">
          <param name="LevelMin" value="INFO"/>
          </filter>
```

4. Locate the `com.vasoftware` package category that looks like this:
```
<category name="com.vasoftware" additivity="false">
          <!-- To turn on the debugging output into vamessages.log,
          change this INFO line to DEBUG and remove LevelRangeFilter of
VAFILE appender -->
          <priority value="INFO"/>
          <appender-ref ref="VAFILE"/>
          </category>
```

5. Set the value of the priority variable to INFO, DEBUG. ERROR, or WARN.

👉   **Note:** This does not require a restart. JBoss picks up these changes on the fly.

Raise the logging visibility of selected database requests

For easier troubleshooting, you can dictate that certain database requests get logged in a handy central log file.

For example, database requests that run longer than 10 seconds are likely candidates for troubleshooting. You can have such requests automatically logged in the `vamessages.log` file for your inspection. The exact length of time after which a request becomes problematic depends on your environment.

How it works:

- All database queries are logged at DEBUG level by default.
- By default, the `vamessages.log` file is configured to include all events logged at the INFO level or higher.
- Database queries that run over a configurable time limit are logged at INFO rather than DEBUG, which causes them to appear in `vamessages.log`.

1. Stop TeamForge.

```
/etc/init.d/httpd stop
/etc/init.d/postgresql-9.0 stop
/etc/init.d/collabnet stop
```

2. Open the `site-options.conf` file.
   This is the master configuration file that controls your TeamForge site.

```
vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
```

☞ **Note:** `vi` is an example. Any *nix text editor will work.

3. In the `site-options.conf` file, change the value of the *LOG_QUERY_TIME_THRESHOLD* variable to a value, in milliseconds, that makes sense for your environment.

4. Run the installer.

   ```
   cd /opt/collabnet/teamforge-installer/6.1.1.0
   ./install.sh -r -I -V
   ```

5. Start TeamForge.

   ```
   /etc/init.d/httpd start
   /etc/init.d/postgresql-9.0 start
   /etc/init.d/collabnet start
   ```

Schedule data extraction for reporting

Set the interval at which you want yourTeamForge site's data extracted to the datamart from which reports are generated.

Each extract-transform-load (ETL) job consists of extracting the data from the production database, transforming it to support reporting, and loading it into the datamart.

By default, this is done every night at 2:30 a.m., by the host's local clock.

1. Open the `site-options.conf` file.

   This is the master configuration file that controls your TeamForge site.

   ```
   vi /opt/collabnet/teamforge-installer/6.1.1.0/conf/site-options.conf
   ```

   ☞ **Note:** `vi` is an example. Any *nix text editor will work.

2. Set the *ETL_JOB_TRIGGER_TIME* variable to the interval at which you want ETL jobs to run.
   For example, a value of `0 0/15 * * * ?` will run an ETL job every 15 minutes.

3. Review the variables you've changed, then save the `site-options.conf` file.

Move datamart to a separate instance

Datamart can be run in same instance or in a separate instance.

1. Do a dump of the PostgreSQL datamart.

   ```
   su - postgres -c 'pg_dumpall -p <datamart-port>
   <dump-location>/<datamart-name>.sql
   ```

2. Stop the services.

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

3. Configure datamart.

   ☞ **Note:** Configuration differs based on the mode of installation. Boostrapping is not required.

4. Restore the reporting database.

   ```
   su - postgres -c dropdb <datamart-name> -p <datamart-port>
   ```

```
su - postgres -c 'createdb -E UTF8 -O <datamart-user> <datamart-name> -p
<datamart-port>

su - postgres -c psql -p <datamart-port> <dump-location>/<datamart-name>.sql'
```

5. Start the services.
6. Check if the existing data is appearing in charts.

Back up and restore CollabNet TeamForge data

CollabNet TeamForge 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:** The items listed in this section address only the data that is either created by or a part of CollabNet TeamForge. 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. Contact your application or operating system vendor for specific guidance on backup strategies for their products.

Back up CollabNet TeamForge 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.

1. Stop TeamForge.

   ☞ **Note:** If TeamForge is running on multiple machines, stop all the machines.

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

2. Run the backup script.

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

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

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

3. Start TeamForge.

   ```
   /etc/init.d/httpd start
   /etc/init.d/postgresql-9.0 start
   /etc/init.d/collabnet start
   ```


Restore backed-up CollabNet TeamForge data

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

1. Stop the CollabNet TeamForge application server.

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

2. Run the restore script.

```
cd /opt/collabnet/teamforge/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 unpacks the backed-up Subversion and CVS repositories, the data directory, and the CollabNet TeamForge database.

3. Restart CollabNet TeamForge .

```
/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 ctfdb > ctfdb.tar
```

☞ **Note:** For this example, the name of the CollabNet TeamForge database is assumed to be ctfdb.

☞ **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 .

3. Create a database and user with the names used for CollabNet TeamForge .

4. Restore the database.

```
createuser -U $CTFUSER createdb -E UNICODE -U $CTFUSER ctfdb pg_restore -d
ctfdb
            ctfdb.tar
```

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

☞ **Tip:** It may also be necessary to restore ownership of the restored tables to the `ctfuser` database user. Something like the following will work (again, assuming the database is called `ctfdb`):

```
for i in `echo "\d" | psql ctfdb | awk {'print $3'}` do echo "ALTER TABLE
 $i OWNER TO $SFUSER;" | psql ctfdb done
```

See the PostgreSQL `pg_restore` man page for more examples and information.

Reference information about TeamForge 6.1.1 on Red Hat

Use this reference information to get deeper detail on configuration files, logs, scripts and other resourcs you use to administrate TeamForge.

Platform specification for TeamForge 6.1.1

This is the hardware and software platform that TeamForge 6.1.1 runs on.

Hardware requirements for CollabNet TeamForge 6.1.1

The following hardware is the minimum recommended for the server on which CollabNet TeamForge 6.1.1 is installed and the server on which the database is installed.

CollabNet TeamForge requires the following minimum server characteristics.

- 2 x CPU 2GHz
- 4 GB RAM (but 16 GB is good for large sites).
- 40GB hard drive

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

> **Tip:** Disk I/O makes a difference. Users in a variety of environments have reported that a high-performance disk subsystem improves the site's response more than additional RAM.

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

Supported software for CollabNet TeamForge 6.1.1

This is the official list of software that is compatible with CollabNet TeamForge 6.1.1.

Operating systems

CollabNet TeamForge has been tested on these operating systems:

Operating system	Version	Architecture	Restrictions
Red Hat Enterprise Linux	5.6	32-bit	Must have RHN or equivalent.
Red Hat Enterprise Linux	5.6	64-bit	Must have RHN or equivalent.
CentOS	5.6	32-bit	
CentOS	5.6	64-bit	
SuSE Linux Enterprise Server	11 SP1	32-bit	Must be registered with Novell.
SuSE Linux Enterprise Server	11 SP1	64-bit	Must be registered with Novell.

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

Package managers

- The yum package manager is supported up to and including version 3.2.1.9.

- The `zypper` package manager is supported up to and including version 1.3.7.

Databases

These database products have been tested with CollabNet TeamForge:

- PostgreSQL 9.0
- Oracle 11 (R1 and R2)
- Oracle Client 11g

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

Browsers

These browsers have been tested with CollabNet TeamForge:

- Mozilla Firefox 3
- Microsoft Internet Explorer 7, 8

Software configuration management tools

These software configuration management (SCM) tools have been tested with TeamForge 6.1.1:

- Subversion 1.6.17 (FSFS)
- CVS 1.11.x
- ViewVC 1.1.10

Java

This Java infrastructure has been tested with TeamForge 6.1.1:

- Java SDK 1.6.0_26
- JBoss Application Server 6.06.0
- Tomcat 7.0.16

☞ **Note:** For larger installs, the max JVM heap size should be increased to 1024MB (or larger), depending on available resources on the box.

Microsoft applications

These Microsoft applications have been tested with CollabNet TeamForge:

- Microsoft Project 2002 (with Service Pack 1) on WinXP Service Pack 2 and Win2000 Service Pack 4
- Microsoft Project 2003 (with Service Pack 1) on WinXP Service Pack 2 and Win2000 Service Pack 4
- Microsoft Office XP (with Service Pack 3) on WinXP Service Pack 2 and Win2000 Service Pack 4
- Microsoft Office 2003 (with Service Pack 1) on WinXP Service Pack 2 and Win2000 Service Pack 4

LDAP (Lightweight Directory Access Protocol) application

OpenLDAP 2.3.27-5 has been tested with CollabNet TeamForge.

Event handling

- Quartz (scheduling tool for synchronous event handling): version 1.8.3

Scripts installed with TeamForge 6.1.1 on Red Hat

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 /opt/collabnet/teamforge
```

Example

The following command forces a bootstrap of the data, showing all actions on the screen:

```
  ./bootstrap-data.sh -n -F -V -d /opt/collabnet/teamforge
```

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 `[DATA_DIR]` is located. You can use the -F option to force bootstrapping on sites that have an existing `[DATA_DIR]`.
-F | --force	This option is only valid when the -n option is used. This option forces the bootstrapping of data when a `[DATA_DIR]` exists.
-V | --verbose	Writes all script actions to the screen. Without this option the script runs silently and logs messages to the `[log_file_directory]/runtime/bootstrap.log` file.
-q | --quiet	Do not show script output.
-f | --site-options=[filename]	Points to the `site-options.conf` configuration file for the site. This argument is optional.

Cluster location

This script runs on the application server machine.

bootstrap-reporting-data.sh

The `bootstrap-reporting-data.sh` script prepares the datamart data for new installations.

Overview

👉 **Important:** This script is only for new datamart installations. If you run it on a site that already has reporting data, all data will be wiped.

The success and error messages from this script are written to the `[log_file_directory]/runtime/bootstrap.log` file.

Usage

Run this script as follows:

```
cd /opt/collabnet/teamforge/runtime/scripts
./bootstrap-reporting-data.sh
```

Options

The following options are available for the `bootstrap-data.sh` script:

-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 `[REPORTS_DATA_DIR]` is located. You can use the -F option to force bootstrapping on sites that have an existing `[REPORTS_DATA_DIR]`.
-F | --force	This option is only valid when the -n option is used. This option forces the bootstrapping of data when a `[DATA_DIR]` exists.
-V | --verbose	Writes all script actions to the screen. Without this option the script runs silently and logs messages to the `[log_file_directory]/runtime/bootstrap.log` file.
-q | --quiet	Do not show script output.

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:

```
/opt/collabnet/teamforge/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 specifed 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`)

datamart-oracle-setup.sh

This script is used for setting up the Oracle datamart in CTF advanced mode installation.

Usage

You can run this script from the command line as follows:

```
[RUNTIME_DIR]/scripts/datamart-oracle-setup
```

The datamart-oracle-setup.sh is an interactive script and does not take any parameters.

Comment

[SCRIPTS_DIR]/set-reports-readonly-user-permission.py script needs to be executed after bootstrap or migration to complete the datamart setup.

datamart-pgsql-setup.sh

This script is used for setting up the PostgreSQL datamart in CTF advanced mode installation.

Usage

You can run this script from the command line as follows:

```
sudo [RUNTIME_DIR]/scripts/datamart-pgsql-setup
```

The datamart-pgsql-setup.sh is an interactive script and does not take any parameters.

Comment

[SCRIPTS_DIR]/set-reports-readonly-user-permission.py script needs to be executed after bootstrap or migration to complete the datamart setup.

db.py

The db.py script can be used to dump and restore a PostgreSQL database.

Overview

This script can be used only for the PostgreSQL service. Don't run this script on a remote database. Execute the script only when the database is up and running.

Usage

Run this script as follows:

```
./db.py --action=<action> --path=<destination directory>
```

Options

Required options:

-a | --action	Values: dump or restore
-f | --path	Path where the database backup file will be created. Must be a directory owned by the postgresql user (usually /var/lib/pgsql/9.0/). Can be a new directory.

Optional options:

-t | --type	Specifies the type of database (ctf or reporting).
-h | --help	Print this usage message and exit.

☞ **Note:** The options -a and -f are mandatory.

domain_change_db.pl

The domain_change_db.pl script handles all the steps required to change the domain name in the site database. It does not change anything in the file system.

☞ **Note:** Changing the domain through any other mechanism may cause problems.

Usage

Execute this script with a command like this:

```
[RUNTIME_DIR]/domain_change_db.pl [--debug] [--dir] --old={domain_name}
--new={domain_name}
```

☞ **Note:** The new domain name must match the value defined for the DOMAIN token in the site-options.conf file.

Options

The domain_change_db.pl script provides the following parameters:

--help	Show this message

--debug	Include debugging output
--old	Old domain
--new	New domain
--dir	Run domain change in this directory only. You must specify the full path.

Use this feature to do a subset of the data directory. This instructs the script to do a recurse in the specified directory looking for the old domain_name and replacing it with the new domain_name..

☞ **Note:** Without this option, only HTML, text, and VM files are modified.

domain_change_fs.pl

The domain_change_fs.pl script handles all the steps required to change the domain name under the file system data directory. It does not change database contents.

Overview

By default, only .s?html?(.[a-z][a-z])?(,v)?, .txt(,v)?, and .vm(,v)? files are modified, unless --dir is specified.

☞ **Note:** Changing the domain through any other mechanism may cause problems.

Usage

Execute this script with a command like this:

```
[RUNTIME_DIR]/domain_change_fs.pl [--debug] [--dir] --old={domain_name}
--new={domain_name}
```

☞ **Note:** The new domain name must match the value defined for the DOMAIN token in the site-options.conf file.

Options

The domain_change_fs.pl script provides the following parameters:

--help	Show this message
--debug	Include debugging output
--old	Old domain
--new	New domain
--dir	Run domain change in this directory only. You must specify the full path.

Use this feature to do a subset of the data directory. This instructs the script to do a recurse in the specified directory looking for the old domain_name and replacing it with the new domain_name..

☞ **Note:** Without this option, only HTML, text, and VM files are modified.

domain_change_pt.py

THIS IS A STUB FILE, TO BE REPLACED BY REAL INFORMATION ABOUT DOMAIN_CHANGE_PT.PY

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.

You will be prompted to run the `install-missing-packages.sh` script if one or more required packages are missing from your system.

☞ **Note:** You may be prompted to manually remove any older packages identified on your system before running the `install-missing-packages.sh` script.

Usage

From the <INSTALLATION_SOURCE> directory, run this script as follows:

```
./environment_check.sh
```

etl-Client.py

The etl-Client.py script allows you to access the Extract, Transform and Load (ETL) scheduler and check the status of the jobs configured. The script also supports triggering jobs manually.

Parameters

The following parameters are available for the etl-Client.py script:

-s | --status	Prints the status of all the jobs configured in ETL service.
-v| --verbose	Chronicles the process of requested operation a bit more.
-r| --run=	Triggers a job manually for a given job.

install.sh

The `install.sh` script handles all operations related to installing and removing the application.

Options
-a | --all

Performs these operations in sequence - Install, create runtime and setup initscripts (equivalent to: -I -r -i)

-b | --bootstrap

Sets up initial site data.

-C | --cleanup

Stop/Kill the application processes, wipes out application packages and site directory.

⚠ **Caution:** THIS OPTION WILL WIPE OUT THE SITE DATA.

-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 the `site-options.conf`
>   file.

-D | --debug

>   Displays traceback errors if any.

-d | --directory type='str', argname='installation_base_dir',

>   Path where the site would get installed.

-E | --check_environment

>   Check if the system environment is suitable for this installation.

-F | --force

>   Force the operation that is performed wherever appropriate (e.g. install/uninstall).

-f | --siteoptionsfile

>   Path to the site configuration file (default: ./conf/site-options.conf).

-I | --initscript

>   Start application services on reboot.

-i | --install

>   Install application packages.

-n | --noninteractive

>   Used to run the installer in non-interactive mode.

-R | --internalruntime

>   Create the internal runtime instance using the site configuration file.

-r | --runtime

>   Create the runtime instance using the site configuration file. (Also does configuration for Apache and
>   PostgreSQL.).

-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.

-u | --uninstall

>   Uninstall application packages.

-V | --verbose

>   Show all output in noninteractive 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 `https://$external_host/cubit_api/1`. 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 `pbl.py` is doing.
`--xml-server-output|--xml`	If this option is not specified, the `pbl.py` 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 `pbl.py` function caches the value of the `--api-user` and `--api-key` parameters in your home directory, in a subdirectory named `.cubit`, the first time a successful authentication is performed against the server. This is analogous to the Subversion client's use of the `.subversion` directory to store authentication credentials. Selecting the `--no-auth-cache` 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 `pub` (the file is in the public area of the PBL) or `priv` (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.

set-reports-readonly-user-permission.py

This script grants read-only access to the datamart for those users specified by the REPORTS_DATABASE_READ_ONLY_USER token.

☞ **Note:** This script is executed automatically when runtime is created, after bootstrap or migration is completed.

Usage

You can run this script from the command line as follows:

```
sudo runtime/scripts/set-reports-readonly-user-permission.py
```

Comment

You can use this script only in advanced mode.

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:

```
/opt/collabnet/teamforge/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 `upgrade-site.sh` 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 `l`, `r`, `u`, or `f` 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 `upgrade-site.sh` 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.

projecttracker.py

The `projecttracker.py` utility provides a command line interface to control a Project Tracker integration on a TeamForge site.

Overview

You can run this script as the root user to start, stop and get the status of a Project Tracker instance that is integrated into your site.

Usage

Run this script as follows:

```
sudo /etc/init.d/projecttracker [command]
```

For example, use the following command to start the Project Tracker integration:

```
sudo ./projecttracker start
```

Commands

The following commands are available for the `projectracker.py` script.

👉 **Note:** "Catalina" is the code name for the integrated Project Tracker feature.

debug -security	Debug Catalina with a security manager
jpda start	Start Catalina under JPDA debugger
run	Start Catalina in the current window
run -security	Start in the current window with security manager
start	Start Catalina in a separate window
start -security	Start in a separate window with security manager
stop	Stop Catalina
stop -force	Stop Catalina (followed by kill -KILL)
version	What version of tomcat are you running?
restart	Stop and Start Catalina
status	Indicate whether Catalina is running

Log files in TeamForge 6.1.1 on Red Hat

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 <SOURCEFORGE_INSTALL_DIR>/log directory.

boot.log	Lgs 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 access_log. This log is rotated each day, and the files have a date stamp appended to their name, such as localhost_access2004-11-26.log.
server.log	Logs all the activities of the application server, including any exceptions. This log is the best place to begin debugging CollabNet TeamForge 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 -specific actions, including some SQL queries that are sent to the backend database. This log is rotated each time it reaches 100MB in size. When rotated the older files have a number appended to the end, such as `vamessages.log.1` and `vamessages.log.2`.

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 ; 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 <SOURCEFORGE_INSTALL_DIR>/log.

Email logs

Both the CollabNet TeamForge 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 `SOURCEFORGE_INSTALL_DIR/james/james-2.2.0/logs`. This log is overwritten each time Phoenix is (re)started. Phoenix is run as part of the CollabNet TeamForge standalone server init script.

CollabNet TeamForge email is handled by the `JAMES` server. `JAMES` logs all of its activities under `SOURCEFORGE_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 .

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

Search logs

Both the CollabNet TeamForge 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 SOURCEFORGE_INSTALL_DIR/james/james-2.2.0/logs. This log is overwritten each time Phoenix is (re)started.

Phoenix is run as part of the CollabNet TeamForge 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 SOURCEFORGE_INSTALL_DIR/james/james-2.2.0/apps/search/logs. The logs that are created are all named default with the date stamp appended to them (such as default-20041126.log). A new log is created for each date that there is indexing activity.

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

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

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.

etl.log

This file contains information from extract-transform-load runs, including data transformation warnings and errors.

👉 **Note:** Transformation errors do not constitute a failed ETL run. For example, if a corrupt row of data in one of the source tables causes transformation errors, this is treated as a "skipped record" and gets logged.

Configuration files in TeamForge 6.1.1 on Red Hat

Edit these configuration files to get the behavior you want.

site-options.conf

CollabNet TeamForge is controlled by settings in a master configuration file called `site-options.conf`. Some of the most useful configuration settings you can specify in the `site-options.conf` file are described here.

The `site-options.conf` file resides in the installer directory. (The default installer directory is `/opt/collabnet/teamforge-installer/6.1.1.0/conf`.)

When TeamForge starts, it reads and implements the instructions provided as values to the variables in this file.

The basic procedure for configuring your TeamForge site, then, is to edit the `site-options.conf` file, supply a valid value for the variable of interest, save the file, and restart the TeamForge runtime environment.

TeamForge has a number of configuration files that can be used to improve site performance. You can configure the `site-options-small.conf`, `site-options-medium.conf` and `site-options-large.conf` files to suit the needs of small, medium and large systems. Each of these sample configuration files has information about the load it supports and the hardware (CPU and memory) required.

ACTIVITY_LINKS_CUSTOMIZATION
When the *ACTIVITY_LINKS_CUSTOMIZATION* variable is set to true, the TeamForge Activity Chart and the Most Active Projects List do not appear on the main page before the user logs in.

Values

true or false

Default

false

ADMIN_EMAIL

The *ADMIN_EMAIL* variable specifies a valid email address for the site administrator.

The mail account specified must be hosted on a separate server from the TeamForge site server.

The *SYSTEM_EMAIL*, *ADMIN_EMAIL*, and *JAMES_POSTMASTER_EMAIL* variables can specify the same address.

> ☞ **Important:** In TeamForge 6.x, the sender name and address for system-generated emails is taken from the value of the *SYSTEM_EMAIL* variable. Therefore, changing the admin user's full name or email address does not affect the sender details of system-generated emails. This is different from TeamForge 5.x, in which the sender name and address for system-generated emails is derived from the admin user's full name and email address.

Values

Email address specification

Default

root@{__APPLICATION_HOST__}

ALLOW_USERNAME_IN_PASSWORD

The *ALLOW_USERNAME_IN_PASSWORD*, when set to true, allows users to set a password that includes the string that they use for their user name on the site.

Values

true, false

Default

true

APPLICATION_LOG_DIR

The *APPLICATION_LOG_DIR* specifies the directory to which the application writes its log files.

Values

Path specification

Default

/opt/collabnet/teamforge/log/apps

APPROVE_NEW_USER_ACCOUNTS

The *APPROVE_NEW_USER_ACCOUNTS* variable specifies whether a site administrator must approve the requests to join the site.

Values

true or false

Default

true

ARTIFACT_LIST_LIMIT

The *ARTIFACT_LIST_LIMIT* variable specifies the maximum number of artifacts that are displayed from a single tracker in the reports or export lists.

Values

Integer

Default

5000

CLEARCASE_INTEGRATION_ENABLED

The *CLEARCASE_INTEGRATION_ENABLED* variable specifies whether the site supports ClearCase source repositories.

Values

true or false

Default

false

DATABASE_HOST

The *DATABASE_HOST* variable is a special case of the *HOST* variable that specifies the host name of the server where the database is running.

Values

Hostname specification

Default

None

Comments

The database host is specified by adding `database` to an existing *HOST_localhost* property, or adding a *HOST_localhost* property if it is not already there.

Example

This site has the database running on the same box as all other services:

```
HOST_localhost=app cvs subversion database
```

This site has the database running on its own separate box:

```
HOST_localhost=app cvs subversion
HOST_<database_host>=database
```

DATABASE_NAME

The *DATABASE_NAME* variable specifies the name of the site's database.

Values

Alphanumeric string

Default

ctfdb

DATABASE_PASSWORD

The *DATABASE_PASSWORD* variable is the password for the Unix user that is authorized to read from and write to the site's database.

Values

Alphanumeric string

Default

ctfpwd

DATABASE_TYPE

The *DATABASE_TYPE* variable specifies the type of database in which the TeamForge site's data is stored.

Values

postgresql or oracle

Default

postgresql

DATABASE_USERNAME

The *DATABASE_USERNAME* variable specifies the Unix user that is authorized to read from and write to the site's database.

Values

Alphanumeric string

Default

ctfrptuser

Comments

For some advanced operations, you may need to log into the database as the database user. However, under normal conditions only the TeamForge site process itself needs to access the database.

DEDICATED_INSTALL

If the *DEDICATED_INSTALL* variable is set to true, the TeamForge site is installed automatically, with the default configuration and minimal user intervention.

Values

true or false

Default

true

Comment

The dedicated install option is appropriate for TeamForge sites where:

- All services (the TeamForge application, the database, and Subversion) run on a single box.
- No other services run on the TeamForge box.

DEFAULT_LOCALE

The *DEFAULT_LOCALE* variable specifies the language in which automated email messages from the site are generated.

Values

Default

en

DEFAULT_PROJECT_ACCESS

The *DEFAULT_PROJECT_ACCESS* variable specifies the type of access that is assigned to a project when it is created. A project can be private, public, or gated.

Values

private, gated, public

Default

private

DISABLE_CREATE_INTEGRATION_SERVERS

The *DISABLE_CREATE_INTEGRATION_SERVERS* token specifies whether the creation of new SCM integrations is allowed.

Values

true or false

Default

false

Comments

When this token is set to its default value of "false", you can add SCM integration servers to your TeamForge site. Also, the **Discover Subversion Edge Servers** option, which enables you to find and connect to Subversion Edge servers on your LAN, is available.

DISABLE_USER_SELF_CREATION

The *DISABLE_USER_SELF_CREATION* variable restricts users from creating their own accounts on the TeamForge home page.

Values

true or false

Default

true

DISCUSSION_DROP_MIME_TYPES

The *DISCUSSION_DROP_MIME_TYPES* variable allows you to delete the mime types submitted by email that contain arbitrary strings.

Values

image/jpeg,image/jpg,text/xml

Default

Regular expression

Example

DISCUSSION_DROP_MIME_TYPES=image/jpeg,image/jpg,text/xml

Comments

Add one or more mime types to the Drop mime types filter. The presence of any of these mime types in an incoming message (via email)causes its deletion with appropriate notification to the posting user.

☞ **Note:** If a mime type is specified in both the Reject and Drop mime filters, then the Reject mime type filter must take higher precedence than the Drop mime type filter.

DISCUSSION_EMAIL_MONITORING

The *DISCUSSION_EMAIL_MONITORING* variable determines which users can monitor a forum on the site.

Values

Value	Description
0	Allow only forum admins.
1	Users with role permissions.
4	All logged in users.
5	Allow all site users and guests.

Default

1

Example

DISCUSSION_EMAIL_MONITORING=4

Comments

This setting applies to the site as a whole. Project owners can choose to be more restrictive in their own project by selecting a lower value on the project administration page.

DISCUSSION_EMAIL_POSTING
The *DISCUSSION_EMAIL_POSTING* variable determines which users on your site can post to forums by e-mail.

Values

Value	Description
0	Allow only forum admins.
1	Users with roles and permissions.
4	All logged in users.
5	Allow known email addresses only.
6	Allow all site users and guests.

Default

1

Example
DISCUSSION_EMAIL_POSTING=4

Comments

This setting applies to the site as a whole. Project owners can choose to be more restrictive in their own project by selecting a lower value on the project administration page.

DISCUSSION_MAX_ATTACHMENT_SIZE
The *DISCUSSION_MAX_ATTACHMENT_SIZE* variable sets an upper limit to the size of files that users can attach to an email message sent to any discussion forum on the site.

Values

Integer (Megabytes)

Default

blank

Comment

A value of zero or less specifies that there is no limit which is the same as the default behavior without the variable.

DISCUSSION_REJECT_CONTENT
The *DISCUSSION_REJECT_CONTENT* variable allows you to block the discussion messages submitted by email that contain arbitrary strings.

Values

Regular expression

Default

None

Example
DISCUSSION_REJECT_CONTENT=(?s).*word.*,(?s).*spam.*

Comments

Add one or more entries. Each regular expression must match an entire entry. The match of any of these entries in the body or subject of an incoming message (via email) causes its rejection, with appropriate notification to the posting user.

☞ **Note:** The content entry is case sensitive.

DISCUSSION_REJECT_HEADERS
The *DISCUSSION_REJECT_HEADERS* variable allows you to block different headers submitted by email that contain arbitrary strings.

Values

Regular expression

Default

None

Example
DISCUSSION_REJECT_HEADERS=(?s).*headername1:value2.*,(?s).*name2:value2.*

Comments

Add one or more header names. Each regular expression must match an entire header name. The match of any of these headers in an incoming message (via email)causes its rejection, with appropriate notification to the posting user.

DISCUSSION_REJECT_MIME_TYPES
The *DISCUSSION_REJECT_MIME_TYPES* variable allows you to delete the mime types submitted by email that contain arbitrary strings.

Values

Application/PDF,text/xml

Default

Regular expression

Example
DISCUSSION_REJECT_MIME_TYPES=application/pdf,text/xml

Comments

Add one or more mime types to the Reject MIME types filter. The presence of any of these mime types in an incoming message (via email) will cause its deletion with appropriate notification to the posting user.

DISCUSSION_ADD_HEADERS

The *DISCUSSION_ADD_HEADERS* variable allows you to add custom headers to the emails posted in the forum.

Values

You can choose to add or remove headers by specifying the particular information you want to be added or dropped from the header. For example, if you add <#d#> in the Add header field, the URL of that discussion will be added to the header of all the available messages in that discussion.

Default

None

Example

DISCUSSION_ADD_HEADERS=headername1:value1, name2: value2 , post-id:<#n#>, forum-url:<#d#>, message-url:<#m#>, domain:<#h#>, list-name:<#l#>, list-address:<#l#>@<#h#>

Comments

Add one or more header names. The match of any of these headers in an outgoing message (via email) causes its addition with appropriate notification to the posting user.

DISPLAY_TIMEZONE

The *DISPLAY_TIMEZONE*variable specifies the time zone displayed in the site, which may be different from the time zone of the server on which the site is running.

Values

The ID for a timezone can be either a full name such as "America/Los_Angeles", or a custom ID in the form GMT[+|-]hh[[:]mm] such as "GMT-08:00". It can also be in the form of a three letter abbreviation such as "PST".

Default

The default timezone of the JVM, if it is undefined.

DOCUMENT_MAX_FILE_UPLOAD_SIZE

The *DOCUMENT_MAX_FILE_UPLOAD_SIZE* variable sets an upper limit to the size of the documents that can be attached.

Values

Integer (Megabytes)

Default

blank

Comment

A value of zero specifies that there is no limit, which is the same as the default behavior without the variable.

DOMAIN_localhost

The *DOMAIN_localhost* variable links the URL at which users can access the TeamForge site to the hostname of the machine where the site is running.

Values

Host and domain specification

Default

None

Example

DOMAIN_appbox.supervillain.org=worlddomination.supervillain.org

Comments

The *DOMAIN_localhost* variable consists of two options. One option identifies the host name of the machine where the TeamForge application is running, and the other specifies the URL through which users will access the services running on that machine.

- To identify the host machine, replace the localhost portion of the *DOMAIN_localhost* variable with the hostname of the machine where the TeamForge application runs.
- To specify the URL, set the value of the variable to the publicly accessible domain name of the site.

👉 **Note:**

- The localhost portion of the *DOMAIN_localhost* variable must match the localhost portion of the *HOST_localhost* variable.
- If the site has its services distributed on multiple machines, the localhost portion of the variable must match the host to which the app option is assigned.

ENABLE_UI_FOR_CUSTOM_EVENT_HANDLERS

When the *ENABLE_UI_FOR_CUSTOM_EVENT_HANDLERS* variable is set to true, site administrators can use the web interface to add custom event handlers to a site.

Values

true or false

Default

true

ENFORCE_MINIMUM_USERNAME_LENGTH

The *ENFORCE_MINIMUM_USERNAME_LENGTH* variable determines the minimum length that can be set for usernames.

Values

0-31

Default

0

ETL_JOB_THREAD_COUNT

The *ETL_JOB_THREAD_COUNT* variable specifies the number of Extract, Transform and Load (ETL) jobs that can be run simultaneously.

Values

1-100

Default

2

Comments

If you only have a few jobs to be triggered few times a day, then 1 thread is sufficient. If you have tens of thousands of jobs, that needs to be triggered every minute, then you probably require thread counts like 50 or 100 (this depends on the nature of the work that your jobs perform, and your systems resources).

ETL_JOB_TRIGGER_TIME

The *ETL_JOB_TRIGGER_TIME* variable specifies the time and date for recurrent Extract, Transform and Load (ETL)jobs.

Values

Cron expression.

Default

0 0/3 * * * ?

Comments

This variable takes a cron expression for a value, and not an absolute time value. The default value evaluates to 2.30 a.m. local time. For help with cron expressions, see *http://en.wikipedia.org/wiki/Cron*.

ETL_SOAP_SHARED_SECRET

The *ETL_SOAP_SHARED_SECRET* variable enables users to access site-wide reporting data via a SOAP client.

Values

String (possibly encrypted).

Default

mightyetlsoapsecret

FORBIDDEN_PASSWORD

The *FORBIDDEN_PASSWORD* variable restricts specified words from being used as passwords.

Values

Comma-separated strings

Default

None

HELP_AVAILABILITY

The *HELP_AVAILABILITY* variable specifies whether context-sensitive online help is served from a network location or from a copy of the content stored on the TeamForge application server. (Context-sensitive help is what the user sees upon clicking the **Help** link in the TeamForge web UI.)

Values

remote, local

Default

remote

HOST_localhost

The *HOST* variable identifies which of the TeamForge site's services are to run on a given machine.

Values

One or more of: app, database, cvs, subversion, etl, datamart

Default

app database cvs subversion etl datamart

Comments

Multiple applications can be assigned to a host. List the applications in space-separated format.

👉 **Important:**

- app (the main TeamForge application) must be assigned to one host only.
- database (the PostgreSQL or Oracle application) must be assigned to one host only.

Examples

Here are some possible variations on the *HOST_* variable when configured on the box where the main TeamForge application is running.

- This host has been assigned the main application, the database, and the Subversion and CVS source control services:

```
HOST_localhost=app database subversion cvs
```

- This is a two-box setup with a database running on a separate machine:

```
HOST_localhost.example.com=app subversion cvs etl datamart
HOST_mydatabase.mydomain.net=database
```

- This is a three-box setup with Perforce and the TeamForge application running on one machine, the database on another and the reporting services on a third:

```
HOST_localhost=app perforce
HOST_mydatabase.mydomain.net=database
HOST_myreportingbox.mydomain.net=etl datamart
```

- This is a three-box setup with the TeamForge application and reporting engine on one machine, both databases on another machine, and the source control services on a third machine.

```
HOST_localhost=app etl
HOST_mydatabase.mydomain.net=database datamart
```

• This is a four-box setup:

```
HOST_localhost.com=app database
HOST_myreportingbox.mydomain.net=etl datamart
HOST_mycvs.mydomain.net=cvs
```

> 👉 **Note:** Observe that when the source control integrations run on a separate box from the main application, no *HOST_* variable is needed on the application box to point to the source control box. The source control machine needs to know how to find the main application box, but the application box does not need to know where the source code integration box is.

HTTPD_LOG_DIR

The *HTTPD_LOG_DIR* variable specifies the path where information about the activity of the TeamForge site's Apache service is written.

Values

Path specification

Default

{__LOG_DIR__}/httpd

INCLUDE_ORGANIZATION_USER_FIELD

The *INCLUDE_ORGANIZATION_USER_FIELD* variable controls whether the organization entry is displayed while creating a user account.

Values

true or false

Default

true

INITIAL_PASSWORD_CHANGE_ACTIVATION_CODE_TIMEOUT

The *INITIAL_PASSWORD_CHANGE_ACTIVATION_CODE_TIMEOUT* variable sets the duration of the validity of the password ticket.

Values

true or false

Default

true

INTEGRATION_JAVA_OPTS

The *INTEGRATION_JAVA_OPTS* variable specifies the memory settings for the Java virtual machine that supports the site's integrated source control services.

Values

Java specifications

Default

-Xms160m -Xmx160m -server -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/tmp -verbose:gc -XX:+
PrintGCTimeStamps -XX:+PrintGCDetails -Dsun.rmi.dgc.client.gcInterval=600000
-Dsun.rmi.dgc.server.gcInterval=600000

INTEGRATION_LOG_DIR

The *INTEGRATION_LOG_DIR* variable specifies the path where information about the activity of the TeamForge site's
source code integrations is written.

Values

Path specification

Default

`{__LOG_DIR__}/integration`

JAMES_GATEWAY_HOST

The *JAMES_GATEWAY_HOST* variable specifies a mail server with Internet access, separate from the TeamForge
server.

Values

Email address specification

Default

None

Comments

Specifying a gateway host 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.

The mail account specified must be hosted on a separate server from the TeamForge site server.

The *SYSTEM_EMAIL*, *ADMIN_EMAIL*, and *JAMES_POSTMASTER_EMAIL* variables can specify the same address.

☞ **Note:** Specify the gateway host by its fully qualified domain name, not a host name.

JAMES_LOG_DIR

The *JAMES_LOG_DIR* variable specifies the path where information about the activity of the TeamForge site's email
component is written.

Values

Path specification

Default

`{__LOG_DIR__}/james`

JAMES_POSTMASTER_EMAIL

The *JAMES_POSTMASTER_EMAIL* variable specifies a valid email address for the person or machine that handles email for the domain, such as postmaster@supervillain.org.

Values

Email address specification

Default

root@{__APPLICATION_HOST__}

Comments

The mail account specified must be hosted on a separate server from the TeamForge site server.

The *SYSTEM_EMAIL*, *ADMIN_EMAIL*, and *JAMES_POSTMASTER_EMAIL* variables can specify the same address.

JAVA_HOME

The *JAVA_HOME* variable specifies the path where the JBoss JVM is running.

Values

Path specification

Default

```
/usr/java/jdk1.5.0_12
```

JBOSS_JAVA_OPTS

The *JBOSS_JAVA_OPTS* variable specifies the memory settings for the JBoss Java virtual machine.

Values

Java specifications

Default

-Xms1024m -Xmx1024m -XX:MaxPermSize=512m -server -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/tmp - verbose:gc -XX:+PrintGCTimeStamps -XX:+PrintGCDetails -Dsun.rmi.dgc.client.gcInterval=600000 -Dsun.rmi.dgc.server. gcInterval=600000

LOGIN_ATTEMPT_LOCK

The *LOGIN_ATTEMPT_LOCK* variable specifies the maximum number of times an user can attempt to access the site.

Values

1-3

Default

3

LOG_DIR

The *LOG_DIR* variable specifies the path where information about the TeamForge site's activity is written.

Values

Path specification

Default

`{__SITE_DIR__}/log`

LOG_QUERY_TIME_THRESHOLD

The *LOG_QUERY_TIME_THRESHOLD* variable enables you to log database requests at INFO level if they run longer than a given period.

By default, database requests are logged at DEBUG level. Configuring a value for *LOG_QUERY_TIME_THRESHOLD* causes requests that run for a period greater than that value to be logged at the INFO level, which makes them show up in `vamessages.log`.

Set the value to zero to log all database queries at INFO.

Values

Integer (in milliseconds)

Default

1000

LOGIN_CONFIG_XML

The *LOGIN_CONFIG_XML* variable specifies the path to the LDAP configuration file.

Values

Path specification

Default

{__DATA_DIR__}/etc/login-config.xml

MAX_WWW_CLIENT

The *MAX_WWW_CLIENT* variable specifies the maximum number of Tomcat request processing threads to be created by the HTTP connector.

Values

Integer

Default

150

MIGRATION_LOG_DIR

The *MIGRATION_LOG_DIR* variable specifies the path where information about the conversion of site data is written during an upgrade.

Values

Path specification

Default

`{__LOG_DIR__}/runtime`

MINIMUM_USERNAME_LENGTH

The *MINIMUM_USERNAME_LENGTH* variable sets the shortest username that the system allows when a user account is created.

Values

Integer (number of characters)

Default

3

MINIMUM_PASSWORD_LENGTH

The *MINIMUM_PASSWORD_LENGTH* variable sets the shortest password that the system allows when a user account is created.

Values

Integer (number of characters)

Default

6

ORGANIZATION_EDITABLE

The *ORGANIZATION_EDITABLE* variable allows or prevents editing the organization value of a user account.

Values

true or false

Default

true

INCLUDE_ORGANIZATION_USER_FIELD

The *INCLUDE_ORGANIZATION_USER_FIELD* variable controls whether the organization entry is displayed while creating a user account.

Values

true or false

Default

true

PASSWORD_REQUIRES_MIXED_CASE
The *PASSWORD_REQUIRES_MIXED_CASE* variable specifies that the user password must contain mixed case letters.

Values

true or false

Default

true

PASSWORD_REQUIRES_NON_ALPHANUM
The *PASSWORD_REQUIRES_NON_ALPHANUM* variable specifies that the user password must contain a non-alphanumeric character.

Values

true or false

Default

```
true
```

PASSWORD_REQUIRES_NUMBER
The *PASSWORD_REQUIRES_NUMBER* variable specifies that the user password must atleast contain one number.

Values

true or false

Default

```
true
```

PERFORCE_LOG_DIR
The *PERFORCE_LOG_DIR* variable specifies the path where information about the activity of the TeamForge site's Perforce source control integration, if any, is written.

Values

Path specification

Default

```
{__LOG_DIR__}/perforce
```

PERFORCE_CLIENT_DIR
The *PERFORCE_CLIENT_DIR* variable specifies the path where the Perforce client is installed.

Values

Path specification

Default

`/usr/local/bin/p4`

PERFORCE_LICENSE_FILE
The *PERFORCE_LICENSE_FILE* variable specifies a path to a file on the server where Perforce is installed, containing the license data for that Perforce installation.

Values

Path specification

Default

`/tmp/license`

PERFORCE_PORT
The *PERFORCE_PORT* variable specifies the port on which Perforce listens for requests.

Values

Port specification

Default

`localhost:1666`

PGSQL_COMMIT_DELAY
The *PGSQL_COMMIT_DELAY* variable specifies the time delay between writing a commit record to the write ahead log (WAL) buffer and flushing the buffer out to disk.

Values

Integer (in microseconds)

Default

`250`

Comments

Together with the *PGSQL_COMMIT_SIBLINGS* option, this option allows a group of otherwise unrelated transactions to be flushed to disk at the same time, with possible significant performance gain.

PGSQL_COMMIT_SIBLINGS
The *PGSQL_COMMIT_SIBLINGS* variable sets the minimum number of concurrent open transactions to require before performing the delay specified by the *PGSQL_COMMIT_DELAY* option.

Values

Integer

Default

`10`

Comments

Together with the *PGSQL_COMMIT_DELAY* option, this option allows a group of otherwise unrelated transactions to be flushed to disk at the same time, with possible significant performance gain.

PGSQL_EFFECTIVE_CACHE_SIZE
The *PGSQL_EFFECTIVE_CACHE_SIZE* variable specifies the size of the OS data cache that is available to PostgreSQL. PostgreSQL can use that data to select the optimal way to execute requests.

Values

Default

Comments

The right value for this variable depends in part on the available RAM on the server where your site is running. Set this value at the highest amount of RAM that you expect to be always available to PostgreSQL.

CollabNet recommends these settings:

8GB RAM	16GB RAM	32GB RAM	64GB RAM	128GB RAM
512MB	1GB	2GB	4GB	8GB

PGSQL_LOG_DIR
The *PGSQL_LOG_DIR* variable specifies the path where information about the activity of the TeamForge site's PostgreSQL database is written.

Values

Path specification

Default

`{__LOG_DIR__}/pgsql`

PGSQL_MAINTENANCE_WORK_MEM
The *PGSQL_MAINTENANCE_WORK_MEM* variable specifies the maximum amount of memory to be used in maintenance operations such as `VACUUM`.

Values

Integer (in MB)

Default

`64`

Comments

For most sites, CollabNet recommends that you set this to 256MB.

PGSQL_MAX_CONNECTIONS

The *PGSQL_MAX_CONNECTIONS* variable determines the number of concurrent connections available to the database server.

Values

Integer

Default

```
100
```

PGSQL_MAX_FSM_PAGES

The *PGSQL_MAX_FSM_PAGES* variable tells the `vacuum` process how many pages to look for in the shared free-space map.

Values

Integer

Default

```
500000
```

Comments

Each FSM page uses 6 bytes of RAM for administrative overhead, so increasing FSM substantially on systems low on RAM may be counter-productive.

PGSQL_MAX_FSM_RELATIONS

The *PGSQL_MAX_FSM_RELATIONS* variable specifies how many relations (tables) will be tracked in the free space map.

Values

Default

```
500
```

PGSQL_MAX_STACK_DEPTH

The *PGSQL_MAX_STACK_DEPTH* variable specifies the maximum safe depth of the server's execution stack.

Values

Integer

Default

```
5120
```

PGSQL_SHARED_BUFFERS

The *PGSQL_SHARED_BUFFERS* variable defines a block of memory that PostgreSQL will use to hold requests that are awaiting attention from the kernel buffer and CPU.

Values

Integer, in MB

Default

240MB

Comments

The right value for this variable depends in part on the available RAM on the server where your site is running. For optimal performance, consider these settings:

8GB RAM	16GB RAM	32GB RAM	64GB RAM	128GB RAM
240MB	480MB	960MB	1.5GB	3.5GB

PGSQL_VACUUM_COST_DELAY

The *PGSQL_VACUUM_COST_DELAY* variable controls the length of time that an I/O process will sleep when the limit set by *vacuum_cost_limit* has been exceeded.

Values

Integer (milliseconds)

Default

50

PGSQL_WAL_BUFFERS

The *PGSQL_WAL_BUFFERS* variable specifies the number of buffers available for the Write Ahead Log.

Values

Default

10MB

Comments

If your database has many write transactions, setting this value bit higher than default may result better usage of disk space.

PGSQL_WORK_MEM

The *PGSQL_WORK_MEM* variable specifies the amount of memory to be used by internal sort operations and hash tables before switching to temporary disk files. .

Values

Integer (in MB)

Default

64MB

Comments

The right value for this variable depends in part on the available RAM on the server where your site is running. For optimal performance, consider these settings:

8GB RAM	16GB RAM	32GB RAM	64GB RAM	128GB RAM
64MB	128MB	256MB	512MB	512MB

PHOENIX_JAVA_OPTS

The *PHOENIX_JAVA_OPTS* variable specifies the memory settings for the Java virtual machine that supports the site's ability to send and receive email and to index data for search.

Values

Java specifications

Default

-Xms256m -Xmx256m -server -XX:+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/tmp -verbose:gc -XX:+PrintGCTimeStamps -XX:+PrintGCDetails -Dsun.rmi.dgc.client.gcInterval=600000 -Dsun.rmi.dgc.server.gcInterval=600000

PROJECT_ACCESS_EDITABLE

The *PROJECT_ACCESS_EDITABLE* variable specifies whether project administrators can change the project access type after the project is created.

☞ **Note:** Site administrators can edit a project's access type regardless of this setting.

Values

true, false

Default

true

REMOTE_HELP_URL

The *REMOTE_HELP_URL* variable specifies the location from which context-sensitive help content is served, if the *HELP_AVAILABILITY* variable is set to remote.

Values

Fully qualified domain name

Default

None

Example

None

REPORTS_DATABASE_NAME

The *REPORTS_DATABASE_NAME* variable specifies the name of the site's reporting database, also known as the datamart.

Values

Alphanumeric string

Default

ctfrptdb

Comments

It is OK for this variable to have the same value as *DATABASE_NAME*, because they are running in separate pgsql processes.

REPORTS_DATABASE_PASSWORD

The *REPORTS_DATABASE_PASSWORD* variable is the password for the Linux user that is authorized to read from and write to the site's reporting database.

Values

Alphanumeric string

Default

ctfrptpwd

Comments

It is OK for this variable to have the same value as *DATABASE_PASSWWORD*, because they are running in separate PostgreSQL processes.

REPORTS_DATABASE_PORT

The *REPORTS_DATABASE_PORT* variable defines a separate port for the reporting database (aka datamart). Using a separate port can improve site performance when database utilization is high.

Values

Port specification

Default

5632

Comments

As of TeamForge 6.1, only port 5632 is supported.

REPORTS_DATABASE_USERNAME

The *REPORTS_DATABASE_USERNAME* variable specifies the Linux user that is authorized to read from and write to the site's reporting database.

Values

Alphanumeric string

Default

ctfuser

Comments

For some advanced operations, you may need to log into the database as the database user. However, under normal conditions only the TeamForge site process itself needs to access the database.

It is OK for this variable to have the same value as *DATABASE_USERNAME*, because they are running in separate PostgreSQL processes.

REPORTS_ENABLE_REPORT_GENERATION

The *REPORTS_ENABLE_REPORT_GENERATION*variable is used to enable or disable the Reports tab in the UI.

Values

true or false

Default

true or false

Comments

Datamart is enabled by adding the 'datamart' service to the HOST_<hostname>token. The service is disabled if datamart is not added. The default value of the REPORTS_ENABLE_REPORT_GENERATION token is based on this service.

REQUIRE_PASSWORD_SECURITY

The *REQUIRE_PASSWORD_SECURITY* variable specifies the password policy for security reasons.

Values

true or false.

Default

true

Comment

This variable can be useful when an organization's security policy prohibits users from entering passwords without any restrictions.

REQUIRE_USER_PASSWORD_CHANGE

The *REQUIRE_USER_PASSWORD_CHANGE* variable determines if the user password needs to be changed during the first login instance.

Values

true or false.

Default

true

Comment

Setting a value true makes the new system force users to change password during first login and false otherwise.

RUNTIME_LOG_DIR

The *RUNTIME_LOG_DIR* variable specifies the path where information about the activity of the TeamForge site's runtime environment is written.

Values

Path specification

Default

```
{__LOG_DIR__}/runtime
```

SCM_DEFAULT_SHARED_SECRET

The *SCM_DEFAULT_SHARED_SECRET* variable allows SCM Integrations to securely communicate with the TeamForge app server.

Values

- Alpha-numeric
- Special characters like '~!@#$%^&*'
- 16-24 byte length

Default

The default value is automatically generated during runtime.

SEARCH_LOG_DIR

The *SEARCH_LOG_DIR* variable specifies the path where information about the activity of the TeamForge site's Lucene search component is written.

Values

Path specification

Default

```
{__LOG_DIR__}/james
```

SEARCH_MAX_FILE_SIZE

the *SEARCH_MAX_FILE_SIZE* variable sets an upper limit to the size of files that are indexed for search.

Values

Integer (bytes)

Default

10M

Comment

A value of zero or less specifies that there is no limit, which is the same as the default behavior without the variable.

SEARCH_SUPPRESS_ARCHIVE_SUB_DOCS

The *SEARCH_SUPPRESS_ARCHIVE_SUB_DOCS* variable prevents archive files from being indexed for search.

Archive files include `zip`, `gzip`, `tar`, and similar file types. They also include document files that are stored in archive format, such as `docx` files from Microsoft Word 2007.

Values

true, false

Default

true

SESSION_COOKIES_ONLY

the *SESSION_COOKIES_ONLY* variable restricts the persistence of all cookies to the user's current session.

If *SESSION_COOKIES_ONLY*=true, then all cookies created during the user session expire automatically when the user closes their browser. If it is false, the cookie expires according to the system logic for that particular cookie.

Values

true or false

Default

false

Comment

This variable can be useful when an organization's security policy prohibits cookies that persist across user sessions.

SITE_DIR

The *SITE_DIR* variable specifies the path where the TeamForge application is installed.

Values

Path specification

Default

`/opt/collabnet/teamforge`

Comments

*   Must match the -d | --directory option to the installer.
*   Spaces are not allowed.

SOAP_ANONYMOUS_SHARED_SECRET

The *SOAP_ANONYMOUS_SHARED_SECRET* variable allows users to have an anonymous login to the TeamForge site through SOAP.

Values

String (possibly encrypted)

Default

None

Comment

The variable must be configured to a non-empty value if users need to have an anonymous login to the site through SOAP. A value must be provided if site-wide reporting is enabled.

SOAP_ARTIFACT_LIST_LIMIT

The *SOAP_ARTIFACT_LIST_LIMIT* variable is used to limit the number of artifacts returned via SOAP calls.

Values

Integer

Default

-1: this means that the artifact list retrieved via SOAP is unlimited

Comments

In TeamForge releases earlier than 6.1.1, SOAP calls returned everything that was asked for, and that is the default behavior in Teamforge 6.1.1 as well. However, sites with performance and stability issues (OutOfMemory errors) in returning a large number of artifacts can now limit the number using this token. Changing this value requires a recreate-runtime and thus a site restart.

> ☞ **Important:** Increasing the number of artifacts beyond the optimal 20,000 - 25,000 range might cause a heap dump.

SSL

The *SSL* variable activates Secure Socket Layer encryption for the TeamForge site.

Values

on or off

Default

on

SSL_CERT_FILE

The *SSL_CERT_FILE* specifies the path to the file where the TeamForge site's Secure Socket Layer certificate is stored.

Values

Path specification

Default

None

SSL_KEY_FILE

The *SSL_KEY_FILE* specifies the path to the file where the TeamForge site's RSA private key is stored when Secure Socket Layer encryption is in effect.

Values

Path specification

Default

None

SUBVERSION_BRANDING_URI

The *SUBVERSION_BRANDING_URI* variable specifies the path component of the data repository URL.

Values

BDB or FSFS

Default

BDB

SUBVERSION_REPOSITORY_BASE
The *SUBVERSION_REPOSITORY_BASE* variable specifies the path to the root directory for the site's Subversion repositories. You can use this variable to put your source code repositories at a custom location on your site's server or on the network.

Values

Path specification

Default

/svnroot

SYSTEM_EMAIL
The *SYSTEM_EMAIL* variable specifies a valid email address for the system administrator responsible for this site.

System administrators can use this email address to set up outage alerts and other notifications.

The mail account specified must be hosted on a separate server from the TeamForge site server.

The *SYSTEM_EMAIL*, *ADMIN_EMAIL*, and *JAMES_POSTMASTER_EMAIL* variables can specify the same address.

👉 **Important:** In TeamForge 6.x, the sender name and address for system-generated emails is taken from the value of the *SYSTEM_EMAIL* variable. Therefore, changing the admin user's full name or email address does not affect the sender details of system-generated emails. This is different from TeamForge 5.x, in which the sender name and address for system-generated emails is derived from the admin user's full name and email address.

Values

Email address specification

Default
root@{__APPLICATION_HOST__}

USE_EXTERNAL_USER_AUTHENTICATION
The *USE_EXTERNAL_USER_AUTHENTICATION* variable specifies whether users can authenticate through a separate system, such as OpenLDAP 2.3.27-5.

Values

true or false

Default

false

USER_ACCOUNT_RESTRICTED
The *USER_ACCOUNT_RESTRICTED* variable determines whether newly created users are "restricted" or "unrestricted" users by default.

- Restricted users can access only public projects and projects of which they are members.
- Unrestricted users can access all projects except private projects of which they are not members.

Values

true or false

Default

true

USER_NEED_PERMISSION_TO_VIEW_FULL_USER_DETAILS
The *USER_NEED_PERMISSION_TO_VIEW_FULL_USER_DETAILS* variable restricts users from viewing other users'
organization information.

Values

true or false

Default

false

USERS_WITH_NO_EXPIRY_PASSWORD
The *USERS_WITH_NO_EXPIRY_PASSWORD* variable specifies the users for whom there is no expiry of password.

Values

Specify the usernames (for the user accounts) for which there is no expiry of password.

Default

admin,nobody,system,cn_monitor (for SaaS) admin,nobody,system (for BTF)

c6migrate.conf variables

These are the configuration settings you can specify in the `c6migrate.conf` file.

Name	Description	Default	Values	Comm
LOG_LEVEL	When set to "INFO", the log is not verbose.	INFO	DEBUG, INFO, WARN, ERROR	
CEE_SITE_DIR	The path to the CEE site directory, for example, `/u1/sourcecast`.			
PROJECT_TRACKER_XML_FILE	The path to the PT XML file, for example, `/u3/PTInstaller-1.0.45/installer/conf/pt.xml`.			
PROJECT_CREATION_DATE	This fall-back date is used when a project's or role's creation/modification date is not found in the audit log.	2000-01-01\ 00:00:00		

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 \bMSI[E] 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} ^/ce-soap50/services/ScmListener
RewriteRule ^/ce-soap50/(.*) http://localhost:8080/ce-soap50/$1 [P]
#Pass all non-listeners SOAP requests. Delete next 4 lines if you don't use SOAP
 APIs.
RewriteCond %{REQUEST_URI} !^/ce-soap50/services/[^/]*Listener
RewriteRule ^/ce-soap50/(.*) http://localhost:8080/ce-soap50/$1 [P]
RewriteRule ^/ce-soap5042/(.*) http://localhost:8080/ce-soap5042/$1 [P]
RewriteRule ^/ce-soap5043/(.*) http://localhost:8080/ce-soap5043/$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
##
```

pebble.app.xml

The `pebble.app.xml` file, also known as the Pebble application configuration file, contains the text that the Pebble application displays in the TeamForge user interface.

This is an example of a default (unedited) `pebble.app.xml` file. To create your own integrated application config file, copy this one into a new file and replace the values with the values appropriate for the application you are integrating.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE integrated-application
  PUBLIC "-//CollabNet, Inc.//DTD Integrated Application Descriptor 1.0//EN"

"http://schema.open.collab.net/sfee50/dtd/sf-pluggable-application-descriptor_1_0.dtd">
<integrated-application>
    <name>Pebble Blog</name>
    <description>l10n.application.description</description>
    <permissions>
        <permission dapMappedTo="View">Blog Reader</permission>
        <permission>Blog Contributer</permission>
        <permission>Blog Publisher</permission>
        <permission>Blog Owner</permission>
    </permissions>
    <prefix>PB</prefix>
    <id-pattern></id-pattern>
    <require-per-project-prefix>true</require-per-project-prefix>
    <require-scm-integration>true</require-scm-integration>
    <!-- Page components for Integrated apps is not implemented for Alpha -->
    <page-component>
      <require-page-component>true</require-page-component>
      <page-component-details>
        <inputtype>text</inputtype>
        <resultformat>html</resultformat>
        <description>l10n.pce.description</description>
        <title>l10n.pce.title</title>
      </page-component-details>
    </page-component>
    <config-parameters>
        <!-- Pebble Configuration Parameters -->
        <param>
            <title>l10n.blogname.title</title>
            <name>blogName</name>
            <description>l10n.blogname.description</description>
            <defaultvalue>My Blog</defaultvalue>
            <displaytype valuetype="String" maxlength="25">TEXT</displaytype>
            <editable>false</editable>
        </param>
        <param>
            <title>l10n.blogdescription.title</title>
            <name>blogDescription</name>
            <description>l10n.blogdescription.description</description>
            <defaultvalue>My Awesome Blog</defaultvalue>
            <displaytype valuetype="String" maxlength="40">TEXT</displaytype>
            <editable>true</editable>
        </param>
        <param>
            <title>l10n.richtexteditor.title</title>
            <name>richTextEditorEnabled</name>
            <description>l10n.richtexteditor.description</description>
            <defaultvalue>checked</defaultvalue>
```

```
                <displaytype valuetype="String">CHECKBOX</displaytype>
                <editable>true</editable>
        </param>
        <param>
            <title>l10n.noofrecentblogentries.title</title>
            <name>recentBlogEntries</name>
            <description>l10n.noofrecentblogentries.description</description>
            <defaultvalue>3</defaultvalue>
            <displaytype valuetype="String">SELECT</displaytype>
            <option name="3">l10n.three.value</option>
            <option name="5">l10n.five.value</option>
            <option name="7">l10n.seven.value</option>
            <option name="9">l10n.nine.value</option>
            <editable>true</editable>
        </param>
    </config-parameters>
    <bundles>
        <bundle locale="en">
            <key name="l10n.application.description">Pebble Blog App</key>
            <key name="l10n.pce.description">Display Blog Title for Given
Date.</key>
            <key name="l10n.pce.title">Enter Blog Date (in yyyy-mm-dd)</key>
            <key name="l10n.blogname.title">Blog Name</key>
            <key name="l10n.blogname.description">Please provide a name for the
 Blog. This appears on all blog pages</key>
            <key name="l10n.blogdescription.title">Blog Description</key>
            <key name="l10n.blogdescription.description">Please provide a
description for the Blog. This appears below blog name on all pages</key>
            <key name="l10n.richtexteditor.title">Rich Text Editor</key>
            <key name="l10n.richtexteditor.description">Enable Rich Text Editor
 for comments and Blog entries?</key>
            <key name="l10n.noofrecentblogentries.title">Recent Blog Entries</key>

            <key name="l10n.noofrecentblogentries.description">How many recent
blog entries do you want to see in the home page?</key>
            <key name="l10n.three.value">3</key>
            <key name="l10n.five.value">5</key>
            <key name="l10n.seven.value">7</key>
            <key name="l10n.nine.value">9</key>
        </bundle>
    </bundles>
</integrated-application>
```

pebble.dep.xml

The `pebble.dep.xml` file, also known as the Pebble deployment configuration file, contains the data that Pebble needs to interact with the TeamForge site.

This is an example of a default (unedited) `pebble.dep.xml` file.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE integrated-application
  PUBLIC "-//CollabNet, Inc.//DTD Integrated Application Descriptor 1.0//EN"

"http://schema.open.collab.net/sfee50/dtd/sf-pluggable-deploy-descriptor_1_0.dtd">
<integrated-application>
    <name>Pebble Blog</name>

<baseurl>{__protocol__}://{__domain__}:{__tomcat.port__}/pebble/index.jsp</baseurl>
```

```
<gourl>{__protocol__}://{__domain__}:{__tomcat.port__}/pebble/gourl/%p/%o</gourl>


<endpoint>{__protocol__}://{__domain__}:{__tomcat.port__}/pebble/services/PebbleIntegrationService</endpoint>
</integrated-application>
```

How is an integrated application described?

An integrated application is described using two XML files - a deployment configuration file and an application configuration file - that provide information to TeamForge about the configuration options exposed by the application.

In TeamForge 6.1.1, you have the ability to configure some integrated application settings using the user interface. You can also export these settings in XML format and make changes. To edit configuration settings, you would upload the XML file containing the updates.

Integrated application settings

👉 **Note:** Some of the tags are internationalized so that the application will display languages based on browser locale. See *Internationalize your integrated application* for more information.

<name>

This is the title of the integrated application. When the integrated application is added to a project, the button that appears on the project pages has this name. This name must be unique -- you cannot use it for any other integrated application on the same TeamForge server.

<description>

This is an internationalized string for the integrated application's description. It contains information for TeamForge project and site administrators to know what the application does.

<baseurl>

This is the URL to which a user will be directed on clicking the integrated application button in a project.

<adminurl>

When an application has an administration screen for configuring its parameters, this field contains that URL. It is optional.

<permissions>

This is a collection of permissions that are exposed by the integrated application. There could be any number of such permissions. These permissions will appear as a part of the project's roles (existing ones, as well as ones newly added) and can be assigned along with other tool permissions. You can map one of these permissions with a "dapMappedTo" attribute -- this indicates the permission to be used when a user logs in without authentication (for example, for public projects). Typically, this is the permission to read data so that it doesn't need a login name; it varies from one application to another.

<require-perprojectprefix>

An integrated application can indicate to TeamForge whether the object ids that it generates are uniquely identifiable across the entire application (if yes, the value for the attribute is "false") or whether they need to be project-specific (in this case, the value for the attribute is "true"). If an integrated application needs per-project prefix, you must enter the prefix value when the integrated application is added to a project.

\<prefix\>

If the "require-per-project-prefix" attribute is false, the value of this tag is used for identifying the integrated application in Go URLs, associations, and linkifications. If the "require-per-project-prefix" attribute is true, the value is used only for the "Host" project. Each project must fill in its value as part of adding the integrated application. (Refer to the doc on how to add an integrated application to a project)

\<gourl\>

This indicates which URL must be used when an object id for an integrated application is specified (either via Jump_to_id or on the URL as /sf/go/\<objectid\>). This URL can support a couple of dynamic parameters.

- `%o` -- The object id entered by the user will be dynamically replaced here
- `%p` -- The project id for the object entered will be dynamically replaced here.

For example, if the Go URL is `http://go.tourl.com/tracking?id=%o` and the object ID entered is `XYZ123`, then the URL will be replaced and redirected to `http://go.tourl.com/tracking?id=XYZ123`.

\<endpoint\>

This is the SOAP endpoint for the integrated application. The endpoint contains the various methods exposed by the integrated application that are called during the lifecycle of TeamForge.

\<require-scm-integration\>

This indicates whether SCM commits need to be vaildated. Some applications might have business rules which indicate that a commit can be made only if certain conditions are met. If the integrated application has any such rules, the value for the attribute should be "true". There are also a couple of methods to be implemented in the SOAP endpoint.

\<page-component\>

These settings are used for Project Content Editors. The integrated application content can become part of the standard Page Component data that appears in project home pages. The settings indicate the type of information that will be available from the integration application.

\<input-type\>	This is the input type control for an integrated app Page Component. We only support 2 types now. Either "select" so that the inputs can be shown from a "SELECT" dropdown and the users will be able to pick a value from there. Else, it can be a "text" where a simple "text" field will be entered for taking the user input.
\<result-format\>	This is the format in which the output of Page Component is returned. This can be a "list" which indicates that it will be a Table like output. The integrated app will send the results in an XML format and the Integrated app framework converts this into a list of records. The other option is "html", where the output from the Integrated application is just displayed on the screen.
\<page-component-description\>	The description that will appear when you add an Integrated application Page Component (Link to the page where " Add component" is available)
\<page-component-title\>	The title that will appear when adding an Integrated application Page Component (Link to the page where "PCE Add component" is available)

\<config-parameters\>

There can be any number of configuration parameters for an integrated application and they are displayed when associating the application to a project. These parameters are filled in by the project administrator and are available in the integrated application SOAP interface as configuration parameters. The integrated application gets a chance to validate these parameters and indicate back to TeamForge whether this project association is successful by passing in a "TRUE". It can return a "FALSE" if it doesn't want this project association to succeed. Each configuration parameter is placed inside the "param" tag, which can contain multiple elements to describe the parameter.

\<title\>	The internationalized title that appears for a project administrator to fill in while associating the integrated application to a project.

\<name\>	The Java variable under which the value for this parameter will be available on the integrated application.
\<description\>	The internationalized description that appears when a project administrator fills in or enters a configuration parameter.
\<default value\>	The default value for the parameter that will appear in the user interface during the association of an integrated application to a project.
\<display type\>	This is the type of display control used for the configuration parameter. We support "TEXT" for text fields, "CHECKBOX" for checkbox type controls, "RADIO" for radio buttons, and "SELECT" for select dropdowns. This field can also take an attribute that says what the value type for the field should be -- whether it should be an "Integer", "String" and so on. So if the field is expecting numbers, then entering "foo" as a value will throw a validation failure.
\<option\>	If the display type is "RADIO" or "SELECT", then these fields contain the individual options available for the display controls. This will contain a "name" attribute that will be sent to the integrated application when that option is selected from the UI. The value of this option should be an internationalized field as it is the value visible to the user.
\<editable\>	This specifies whether the configuration parameter should be editable once the integrated application is associated to a project. These configuration parameters are available when you add or edit an integrated project. If a parameter should not be "edited" post association, setting this to "false" will make it non-editable.

\<id-pattern\>

When trying to link to an integrated application id, this regular expression gets used for mapping. By default (if no value is provided), it looks for alphanumerical characters; in case you need specific characters to be matched (for example, JIRA, which has hyphens in ids), this value is used.

\<require-page-component\>

Some integrated applications choose not to expose details as Page Components. For those that don't, set this tag to "false" and for those that do, set it to "true". If the value is "true", you must provide the "page-component-details" tags aw well.

\<service-type\>

TeamForge 6.1.0 and earlier releases supported only SOAP as the mechanism to talk from TeamForge to the integrated application. TeamForge 6.1.1 also supports REST calls. The service-type tag indicates whether the protocol used for communication is REST or SOAP.

For examples of how these tags are used in the integration of the Pebble bogging application, see *pebble.dep.xml* and *pebble.app.xml*

install.conf

The `install.conf` file contains the data needed to manage the Pebble installer.

This is an example of a default (unedited) `install.conf` file. To create your own Pebble installer config file, copy this one into a new file and replace the values with the values appropriate for the application you are integrating.

```
#Location of the Pebble Blog
pebble.base.dir=/u1/pebble
#CTF Information
ctf.baseurl=http://cu073.cubit.maa.collab.net/
#Tomcat Information
tomcat.port=13000
domain=cu073.cubit.maa.collab.net
timezone=Asia/Calcutta
java_home=/usr/java/jdk1.6.0_26/
```

```
protocol=http
java_opts=-Xms512m -Xmx512m
# This where the Pebble Blogs are stored
data.dir=/u1/pebble-data
secretkey=mistywasacat
```

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
```

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 site* on page 68. Option values that must be modified are highlighted in bold.

- When the username is passed to the login module from TeamForge, 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>
```

Sample code for Active Directory integration

Active Directory is not supported. However, these sample lines in the `login-config.xml` file may help you make it work for a simple AD setup, without complex directory structures requiring additional search parameters.

Set the values of `java.naming.provider.url`, `principalDNSuffix` and `rolesCtxDN` as appropriate to your site.

For more detailed instructions, see *http://www.jboss.org/community/wiki/LdapLoginModule*.

```
        <login-module code="org.jboss.security.auth.spi.LdapLoginModule"
flag="required" >
            <module-option
name="java.naming.provider.url">ldaps://<server_name>:636/</module-option>
            <module-option name="allowEmptyPasswords">false</module-option>
          <module-option name="principalDNSuffix">@foo.bar.com</module-option>

        <module-option name="rolesCtxDN">dc=Foo,dc=Bar,dc=Com</module-option>

            <module-option name="matchOnUserDN">true</module-option>
            <module-option name="uidAttributeID">sAMAccountName</module-option>

            <module-option name="roleAttributeID">memberOf</module-option>
            <module-option name="roleAttributeIsDN">true</module-option>
            <module-option name="roleNameAttributeID">name</module-option>
        </login-module>
```

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 (#).

postresql.conf

The `/var/lib/pgsql/9.0/data/postresql.conf` file controls the behavior of the PostreSQL database.

Shared port

- If the database and the datamart are using the same port (port 5432), this configuration file supports both.
- If the database and the datamart are on separate boxes, identical copies of this configuration file must exist in `/var/lib/pgsql/9.0/data` on both boxes.

```
# ---------------------------
# PostgreSQL configuration file
# ---------------------------
#
<snip>

#-----------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#-----------------------------------------------------------------------------

# - Connection Settings -

#listen_addresses = 'localhost'  # what IP address(es) to listen on;
      # comma-separated list of addresses;
      # defaults to 'localhost', '*' = all
      # (change requires restart)

listen_addresses = '127.0.0.1,<database_host_ip>'  # what IP address(es) to
listen on;
#port = 5432 # (change requires restart)


<snip>
```

Separate ports

If the database and the datamart are on the same box but using separate ports, copies of this configuration file must exist in both `/var/lib/pgsql/9.0/data` and `/var/lib/pgsql/9.0/reports`. Each copy must identify a different port.

This file is in `/var/lib/pgsql/9.0/data`.

This copy of the file is in `/var/lib/pgsql/9.0/reports`. Note the different port number.

```
# -----------------------------
# PostgreSQL configuration file
# -----------------------------
#
<snip>

#-----------------------------------------------------------------------------
# CONNECTIONS AND AUTHENTICATION
#-----------------------------------------------------------------------------

# - Connection Settings -

#listen_addresses = 'localhost'  # what IP address(es) to listen on;
      # comma-separated list of addresses;
      # defaults to 'localhost', '*' = all
      # (change requires restart)

listen_addresses = '127.0.0.1,<database_host_ip>'  # what IP address(es) to
listen on;
#port = 5632 # (change requires restart)


<snip>
```

Frequently asked questions about administrating a TeamForge 6.1.1 site on Red Hat

Use this background knowledge to help you install, maintain and support a CollabNet TeamForge site.

General questions about using TeamForge

Common questions about using TeamForge.

How does CollabNet TeamForge compare to CollabNet Enterprise Edition?

You can help create a seamless transition for your site's users from CollabNet Enterprise Edition to CollabNet TeamForge.

blah blah blah

What does it take to install CollabNet TeamForge ?

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

Before you begin the installation you will want to consider hardware requirements and other factors like supported software. For more information about installation see *Install Reference*.

Do I need an advanced TeamForge installation?

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.

Remote boxes

Many sites benefit from running some of the TeamForge services on one or more separate boxes. You can only do this with an advanced installation.

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 your TeamForge site with SSL* on page 81 for information.
- If your site requires SELinux, you must configure your Apache service. See *Set up SELinux* on page 80 for the recommended settings.

In either case, use the advanced install instructions.

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

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

CollabNet TeamForge functionality is delivered by five discrete services. Each service can run on its own machine or share a machine with one or more other services. You assign specific services to specific boxes when you customize your CollabNet TeamForge installation by editing the `site-options.conf` file.

CollabNet TeamForge 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 `svn` 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 `cvs` application, running on an arbitrary number of machines.
etl	The reporting service.
datamart	A mirror of your site database for the reporting engine to work with.

In principle, services can be combined in any configuration, with some constraints, such as:

- Only one site database, datamart and reporting service.
- Multiple source control integration services, but only one per box.

In practice, CollabNet has identified five configurations as the most useful for a wide variety of site. You can follow the instructions here to set up your site in one of these configurations, or you can adapt one of them to your own conditions.

How does TeamForge handle third-party applications?

TeamForge relies on many third-party applications to augment or enhance functionality.

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

CVS

Technical Support provides best-effort support for Subversion and CVS client usage issues. TeamForge is not shipped with CVS source control functionality. For best results, contact the vendor 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 Red Hat for a solution to this bug for any case where users are members of 32 or more CVS repositories on a TeamForge CVS server.

Discussion forum threading

For TeamForge 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.

How does TeamForge manage security?

TeamForge is a secure, centralized, enterprise-grade solution for optimizing distributed development.

A number of factors go into ensuring security, for detailed information about TeamForge security management see *How does CollabNet SourceForge Enterprise 5.0 manage security?*

What are the minimum ports to keep open for a TeamForge site?

The components of a CollabNet TeamForge installation listen on a number of operating system ports. A small subset must be exposed externally to enable users to access TeamForge services. Any port that is not absolutely needed must be closed.

⚠️ **Caution:** Expose only the JBOSS and Tomcat ports that are required for integration with another application, and open them only to that specific host IP address, even within your internal network.

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

- Use the firewall configuration GUI tool that comes with your operating system. It's usually launched with a command like `system-config-selinux`.
- Open the `/etc/sysconfig/iptables` file and specify your open ports by hand.

Ports open to the Internet

Open the following operating system level ports. All other ports must be firewalled off to maintain security.

👉 **Important:** Do not open port 7080 or port 8080 to the Internet. These ports are only for communications between the TeamForge application and the source code integration service, when those two site components are running on separate boxes.

22 (SSH)	Port 22 is the default port for the secure shell (SSH). This is required for basic SSH administrative functionality and for CVS, as all CVS transactions occur over SSH. If all Teamforge repositories are

in SVN (the default for Teamforge), then this port should be closed to the public and only accessible to the system administrators.

If you have to expose SSH to the Internet, the best way to protect it is to require SSH keys and not allow password authentication, and do not permit root logins over SSH. If you must use local authentication for SSH, enforce regular password changes and password complexity.

☞ **Note:**
- If you have to expose SSH internally, limit access to the port to a bastion host if you can; otherwise limit it to specific trusted hosts or subnets.
- Do not expose cvspserver (the TCP protocol over port 2401) either internally or to the Internet if there is any way you can avoid it.

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

80 (HTTP)   Port 80 is the default port for Web data transfer. We strongly recommend that you set up SSL and use port 80 only to redirect to port 443.

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).

TeamForge can help you take care of this, if you tell it to. See *Set up SSL for your TeamForge site* on page 81 for details.

Ports for internal use only

Ports 7080 and 8080 have special internal uses for your site, but should not be exposed externally.

7080   On the source code integration server, if it is a separate physical server from the TeamForge application server, expose a port by which the application server can communicate with the SCM integration server. The default is port 7080.

8080   If you are running the source code (CVS, Subversion, or Perforce) integration server on a separate physical server from the TeamForge application server, set port 8080 on the TeamForge application box to accept connections from the server where your source code integration service is running.

☞ **Important:** Do not open port 7080 or port 8080 to the Internet. These ports are only for communications between the TeamForge application and the source code integration service, when those two site components are running on separate boxes.

How does CollabNet TeamForge 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 application server nor the backend servers should ever be exposed to the Internet.

The CollabNet TeamForge 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 application server.
- Send mail to CollabNet TeamForge 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 for commit notifications.
- The mail server must be able to deliver messages across the firewall.

Authentication and authorization

To secure sensitive data, CollabNet TeamForge 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 .

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 auditing tools are a powerful way to track unwanted and/or unauthorized changes within the system.

How does CollabNet TeamForge 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 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 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.

J2EE Architecture and security

CollabNet TeamForge 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 ?

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

👉 **Note:** Although CollabNet intends CollabNet TeamForge 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 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 requires browsers to support cookies. Cookies are used for the sole purpose of managing user sessions. CollabNet TeamForge 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 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 authenticates a user's login information. A session is invalidated when one of following events occur:

- The user explicitly logs out of CollabNet TeamForge.
- 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 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 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. Similarly, administrators can allow or reject dictionary-words, force passwords to expire, and enforce upper/lower case/special character combinations. Moreover, CollabNet TeamForge administrators can enforce password expiration and other policies.

Cross-site scripting (XSS) protection

CollabNet TeamForge 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 provides.

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

How does TeamForge authenticate CVS users?

CVS is treated as a special case when managed by a TeamForge site. It is not authenticated in the same way as SOAP API clients.

CVS relies on the Linux operating system to provide access and security. This includes permissions on individual repositories and access to the server itself. To add users, change passwords, create repositories, etc., the TeamForge integration simply changes the appropriate settings in the Linux operating system.

Users access CVS via an encrypted SSH session. To support this, TeamForge creates accounts on the Linux server that hosts the CVS repository. A typical CVS repository is created in the `/cvsroot` directory and is owned by root, with a group assigned by TeamForge. To gain access to a repository, TeamForge will add a user to the appropriate system group.

When TeamForge detects that a user's password has changed, it changes the password for that user on the Linux server too.

Users also have the option to use SSH keys or Kerberos tickets.

When a user is added to the Linux server, the login shell is `cvssh`, which limits their activities to CVS commands.

👉 **Important:** Do not expose cvspserver (the TCP protocol over port 2401) either internally or to the Internet if there is any way you can avoid it.

Password changes under internal authentication

To set the password for the user at the operating system level, TeamForge needs to have the clear-text version of the users password. The only time TeamForge has this is when the user uses the Change Password form in the Web UI. This is because the database-stored version, as an MD5 Password Hash, is a one-way encryption and can't be decrypted.

On a successful password update, TeamForge makes a SOAP call to the integration server that manages CVS. For this reason, the integration server must be SSL-enabled.

Password changes under external authentication

When a password change happens in an external authentication system, TeamForge does not immediately know that the password has changed. TeamForge needs a way to detect that the password has changed.

To accomplish this it keeps a copy of the last password the user successfully logged in with as an MD5 Password Hash in the same database table and field that it normally uses for Internal Authentication.

Now that TeamForge has a reference point, it still needs a clear-text copy of the password to make the change at the Linux operating system level. The only time this is available is when the user logs into TeamForge via the Web UI or SOAP API. So upon a successful login TeamForge compares the password to the encrypted one. If it is different it tells the Linux operating system to change the users password and then saves it in the database (as an MD5 Password hash).

Until the user logs into TeamForge, the CVS server will still have and accept the old password. There is no CVS server-side way to trigger a password update, unless an alternative method is used, such as LDAP or Kerberos.

Alternative Authentication
Because users use SSH to access a TeamForge-managed CVS server, it is possible to configure SSH to accept other authentication features such as SSH keys and Kerberos tickets. It is even possible to disable the use of passwords and require the use of other alternative methods.

- TeamForge supports SSH Keys natively. The user uploads their public key into their profile under "My Settings" in the TeamForge Web UI. The key will automatically be copied to each CVS server that TeamForge manages.
- If TeamForge is using External Authentication and the method is Kerberos, then SSH can be configured to use the same Kerberos server. This allows users to use Kerberos tickets for CVS operations.

LDAP
Linux supports LDAP as an authentication source. If TeamForge is using External Authentication and the source is LDAP, then SSH can also use that same source. When you do this, passwords and user account status are observed in real time instead of as a mirror of TeamForge.

How do I configure Subversion to authenticate against multiple LDAP domains?
For some configurations, a Subversion server may need to be authenticated against multiple LDAP domains. This is possible by modifying the Apache configuration.

This is now possible due to the mod_authn_alias module for Apache. The external link for the module contains multiple usage scenarios. You will need to confirm that your Apache has been compiled with the module enabled. (This is the case for CollabNet Subversion binary packages since 1.5.4). If it is compiled as a module, make sure it is enabled via the LoadModule directive in your Apache configuration.

Example configuration usage for authentication against three LDAP servers :
```
<AuthnProviderAlias ldap ldap-US>
    AuthLDAPBindDN cn=ldapuser,o=company
    AuthLDAPBindPassword password
    AuthLDAPURL
ldap://ldap-us.company.local/ou=Developers,o=company?sub?(objectClass=*)
</AuthnProviderAlias>

<AuthnProviderAlias ldap ldap-EU>
    AuthLDAPBindDN cn=ldapuser,o=company
    AuthLDAPBindPassword password
    AuthLDAPURL
ldap://ldap-EU.company.local/ou=Developers,o=company?sub?(objectClass=*)
</AuthnProviderAlias>

<AuthnProviderAlias ldap ldap-IN>
    AuthLDAPBindDN cn=ldapuser,o=company
    AuthLDAPBindPassword password
    AuthLDAPURL
ldap://ldap-in.company.local/ou=Developers,o=company?sub?(objectClass=*)
</AuthnProviderAlias>
```

```
<Location /svn>
    DAV svn
    SVNParentPath /opt/subversion/repos
    AuthType Basic
    AuthName "Subversion Repository"
    AuthBasicProvider ldap-US ldap-EU ldap-IN
    AuthzLDAPAuthoritative off
    Require valid-user
</Location>
```

How do I authenticate multiple LDAP via Apache?

If you need to add multiple OU= values in the LDAP url you must have separate LDAP urls and utilize AuthnProviderAlias to check both LDAP searches.

Use the following AuthnProviderAlias to check LDAP searches.

```
LoadModule authn_alias_module
modules/mod_authn_alias.so

<AuthnProviderAlias ldap ldap-alias1>
AuthLDAPBindDN cn=youruser,o=ctx
AuthLDAPBindPassword yourpassword
AuthLDAPURL ldap://ldap.host/o=ctx
</AuthnProviderAlias>

<AuthnProviderAlias ldap ldap-other-alias>
AuthLDAPBindDN cn=yourotheruser,o=dev
AuthLDAPBindPassword yourotherpassword
AuthLDAPURL ldap://other.ldap.host/o=dev?cn
</AuthnProviderAlias>

Alias /secure /webpages/secure
<Directory /webpages/secure>
Order deny,allow
Allow from all

AuthBasicProvider ldap-other-alias ldap-alias1

AuthType Basic
AuthName LDAP_Protected_Place
AuthzLDAPAuthoritative off
Require valid-user
</Directory>
```

After switching to ADS authentification, why did the Create button disappear from the user admin section?

When using external authentication such as LDAP, creating users from within the application is disabled. All users must be created via LDAP.

Does TeamForge work with LDAP?

Yes, you can have your TeamForge installation authenticate against an LDAP server. This is handy when users want to use a variety of different resources without having to maintain credentials for each one separately.

Overview

CollabNet TeamForge is a JBoss2 based application and relies on the JBoss JAAS service for user authentication. This enables a TeamForge site to authenticate users internally or externally.

Internal user authentication	Out of the box, TeamForge relies on its local database to manage user accounts. This includes username, password, full name, email address and a variety of other meta data values. Passwords are stored in the database using the standard MD5 Password hashing algorithm1.

The database is only accessible by the application itself and a user with root access to the physical server. While running in this default configuration users are allowed to change their passwords in TeamForge, and any user with site administration privileges can create and approve new user accounts.

External user authentication   The JAAS service comes with several standard providers that allow TeamForge to be integrated with services such as LDAP, Active Directory and Kerberos. The JAAS service allows more than one source to be configured in the event several sources are needed.

☞   **Note:** It is possible to use both types of authentication with a single TeamForge installation. See your CollabNet representative for details.

To ensure that you are not locked out of your site, the site administrator account is always validated by TeamForge, not by LDAP.

LDAP accounts must conform to the TeamForge rules for user names and passwords. For example:

- If a password is used in LDAP that is shorter than the minimum allowable password length in TeamForge, you cannot create the user in TeamForge.
- A user name that starts with a special character, such as an underscore, will not be accepted by TeamForge, even if it is valid in LDAP.

(For detailed TeamForge user name and password rules, see *Create a new user account*.)

How is life different for the user under external authentication?

- When you turn external integration on, every user account (except the site administrator account) must have a matching LDAP entry to log in. This may require changing some existing accounts to match their corresponding LDAP records. (Accounts created after LDAP is in place are validated with the LDAP server when they are created, so you don't have to worry about this.)
- Every login attempt (Web UI and SOAP access) is passed to the external provider. This means that any changes to the user status in the external system take effect immediately. Users who have already logged in and have valid sessions are not affected.
- When TeamForge is using internal authentication, a site administrator can change a user's password. This is disabled for external authentication.
- Under external authentication, passwords can't be changed in the TeamForge web UI. Users have to use the interface provided by the third-party authentication source to change their password. Such password changes are available immediately to TeamForge for the next login attempt.
- Site administrators can no longer create user accounts. The end user must create their own account by logging into TeamForge just like a user who already has an account. At that point TeamForge detects that a new account needs to be created and presents the new user with a registration form, which requests the user's password n the external authentication system. On submit, TeamForge verifies the user account with the external system, and only if the username/password is verified does TeamForge create the new account.
- Once a new user has created their account, TeamForge can optionally be configured to put every new account in a pending status so that a site administrator can approve the new account. By default, new users will have immediate access to the system.

LDAP for source control

LDAP is integrated into your TeamForge source control services.

- For Subversion, the integration server queries TeamForge as needed.
- CVS authentication is not managed directly by LDAP, but each TeamForge user's SCM password is synchronized automatically with the user's LDAP password upon logging into TeamForge.

What can go wrong?

When TeamForge is configured to authenticate against an LDAP server and the LDAP server is down, all TeamForge 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 TeamForge.

Why do I get the "Invalid command 'AuthLDAPAuthoritative'" error when I try to set LDAP for SVN users?

The invalid command `AuthLDAPAuthoritative` error may occur if you need to upgrade Apache from version 2.0 to 2.2.

CollabNet Subversion 1.5 is bundled with the latest version of Apache (currently 2.2.x). It includes the module mod_authnz_ldap and does not include mod_auth_ldap. Hence compatibility issues arise due to missing directives. Upgrade your Apache version to 2.2 if you get the following error when trying to install CollabNet SVN:

```
 bash-3.00# /etc/init.d/collabnet_subversion start
    Starting CollabNet Subversion:
    Syntax error on line 29 of
        /etc/opt/CollabNet_Subversion/conf/collabnet_subversion_httpd.conf:
    Invalid command 'AuthLDAPAuthoritative',
        perhaps misspelled or defined by a module not included in the server
configuration
    FAILED
```

How does TeamForge handle multiple redundant LDAP servers?

When configuring LDAP authentication for a TeamForge instance, there may be a business need for using multiple LDAP servers. Follow the guidelines below for configuring.

The additional LDAP servers can be added to the java.naming.provider.url option in login-config.xml:

```
login-config.xml:
        <module-option name="java.naming.provider.url">
        ldap://primary/ ldap://secondary/</module-option>
```

Once the primary and secondary servers have been defined, they will be consulted in order of definition for every authentication request. First the primary, and if the primary fails, then the secondary. This prevents specifying multiple servers for round-robin handling of authentication, but it can still be used for redundancy needs.

Can the users be forced to change their passwords at first login?

Yes, as a site administrator you can configure the CollabNet TeamForge site options to force the users to change their passwords at first login.

Setting the `REQUIRE_USER_PASSWORD_CHANGE` attribute as `true` in the `site-options.conf` file enforces password change on first login into CollabNet TeamForge.

👉 **Tip:** You can not force password change on a user who had self-created the user account, or if a password-request had been raised for the user or if an administrator had reset the login password for that user.

Managing email in TeamForge

Questions about managing email in TeamForge.

How do I configure TeamForge to send mail on a specific network adapter in a multi-NIC configuration?

When a host has multiple NICs, James will try to do the right thing when sending mail. In some network setups, this is not correct, and manual configuration is needed.

James requires multiple changes to fully configure how it interacts with the network. Open the config.xml file, located in `$SF_HOME/apps/james/james-2.2.0/apps/james/SAR-INF/` for version 5.1.

Locate the `'<mailet match="All" class="RemoteDelivery">'` section. add a subnode `'<bind>$addr</bind>'` where $addr is the ip address that James should be sending mail from.

Near that area, there is a <servernames...></servernames> section. Confirm/change the two autodetect options (autodetect, autodetectIP) to false. Next, add the fully qualified host name, and the ip address that will be used, to their own <servername> entry.

After the changes are complete, save the config.xml and restart the application.

How can I check if port 25 is open?

If you know the mail server is up and running, check whether you can talk over port 25 to your mail server. This can be done using a one-line command: telnet <appserver name> 25 Substitute the <appserver name> with your own server.

Once you type this into your DOS window and hit return, you should see some sort of response from your mail server, as shown below:

Trying 208.75.196.84... Connected to cu190.cubit.sp.collab.net (208.75.196.84). Escape character is '^]'. 220 cu190.cubit.sp.collab.net SMTP Server (JAMES SMTP Server 2.2.0) ready Mon, 27 Jul 2009 06:38:20 -0700 (PDT)

How do I set up a local alias via James?

In situations where you need to obtain a SSL certificate for your domain, and your SSL certificate provider only permits you to use addresses related to your TeamForge domain, it may be necessary to generate an email alias from within TeamForge. Since there is currently no way to do this through the UI, you'll have to do it from the James administrative interface.

First, you'll need to connect to the James administrative interface on your system. If you've followed our best practices guide in our knowledgebase, you'll know that you should have port 4555 firewalled to everyone but localhost. SSH to your TeamForge server, and then issue the following command:

```
telnet localhost 4555
```

This will bring up the Remote Administration Tool:

```
[root@app1 root]# telnet localhost 4555
Trying 127.0.0.1... Connected to localhost (127.0.0.1).
Escape character is '^]'. JAMES Remote Administration Tool 2.2.0
Please enter your login and password
Login id: admin Password: (text is echoed locally)
Welcome admin. HELP for a list of commands
```

First, we'll need to add a new user:

```
adduser <username> <password>
```

Then, we'll need to set the forwarding address of that user

```
setforwarding <username> <email address where you want email to go>
```

finally, we'll exit the James administrative interface.

```
quit
```

Your changes should be in place.

How do I configure email notifications of Subversion commits in SourceForge 4.x?

For SourceForge release 4.x, email notifications are not available via the web UI, and will require extra commit hook scripts to be installed.

To implement commit email notifications on SourceForge 4.x, you must install extra commit hook scripts in the repository. For detailed information see the online Subversion documentation: *Hook Scripts*.

☞ **Note:** This feature has been implemented in SourceForge 5.x. In the current release, a user can monitor a repository, much the same way they monitor a tracker for discussion forum.

Does TeamForge support using /etc/aliases for local mail delivery?

No, TeamForge uses the James SMTP server, which does not use the /etc/aliases file.

To enable local mail aliases, you will need to configure user mapping in the XMLVirtualUserTable in the `/opt/collabnet/teamforge/runtime/james/apps/james/SAR-INF/config.xml` file.

☞ **Note:** Please note that while the James SMTP server is used as part of TeamForge, customizations such as these cannot be supported by CollabNet.

Should I upgrade to TeamForge 6.1.1 on a new box?

You can upgrade TeamForge on new hardware, or you can stick with the same box your current site is running on. To choose, consider how your members use your site and how you maintain it.

Upgrade on new hardware

Upgrading on new hardware is a little more complex, but provides more flexibility.

Do it this way if you want to:

- Change to a different operating system.
- Stage your upgraded site before moving it into production.
- Serve the site from a different data center.
- Support an expanded user base.
- Minimize downtime associated with the upgrade.

Upgrade on the same hardware

If you are OK with your current hardware setup and you don't want to rock the boat, you can run your new TeamForge site on the same box where your current site is running.

Do this if you want to:

- Keep your upgrade as simple as possible.
- Stick with a tested hardware setup.
- Use virtual hardware.
- Support a stable user base.

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

What are the right PostgreSQL settings for my site?

Your site's PostgreSQL settings depend on the conditions your site is operating under, especially the number and size of projects and the number of users.

The default values in the `site-options.conf` file are designed for a TeamForge site running on a system with 8 GB of RAM. This table contains recommended values for systems with various amounts of RAM, based on testing carried out in CollabNet's performance lab. Use your discretion in selecting the right values for your environment.

☞ **Note:** Remember to recreate the runtime environment after changing any value in the `site-options.conf` file.

	8GB RAM	16GB RAM	32GB RAM	64GB RAM	128GB RAM
PGSQL_COMMIT_DELAY	250	250	250	250	250
PGSQL_COMMIT_SIBLINGS	10	10	10	10	10
PGSQL_EFFECTIVE_CACHE_SIZE	512MB	1GB	2GB	4GB	8GB
PGSQL_MAINTENANCE_WORK_MEM	256MB	256MB	256MB	256MB	256MB

PGSQL_MAX_FSM_PAGES	500000	500000	500000	500000	500000
PGSQL_MAX_FSM_RELATIONS	500	500	500	500	500
PGSQL_SHARED_BUFFERS	240MB	480MB	960MB	1.5GB	3.5GB
PGSQL_WORK_MEM	64MB	128MB	256MB	512MB	512MB
PGSQL_VACUUM_COST_DELAY	50	50	50	50	50
PGSQL_WAL_BUFFERS	10MB	10MB	10MB	10MB	10MB

Is it possible to change artifact prefixes in TeamForge?

Artifact prefixes in Teamforge cannot be changed or customized, all artifacts across the site will have an artifact ID of artfXXXXXX.

Can I run other java applications in the same JBoss instance as CollabNet Team Forge?

It is recommended to create a separate instance of JBoss or tomcat and deploy your applications there. While it is possible to deploy other applications alongside CTF, you may encounter some errors.

In order to deploy an application into the CTF JBoss instance, you must place the war files into the deploy directory beneath the runtime directory. All directories beneath the runtime directory are recreated each time the application runtime is rebuilt, deleting your applications.

If your applications include the CTF SDK , you may receive class conflicts and errors such as "java.lang.reflect.InvocationTargetException" when attempting to connect to CTF.

Who is responsible for applying OS updates to the underlying VMware image?

CollabNet does not provide the OS updates. If there is a critical update that needs to be addressed within the VM Image, CollabNet will post instructions on how to update that on open.collab.net.

Concepts and terms in TeamForge

Descriptions of concepts and terms in TeamForge.

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

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

To access CollabNet TeamForge , one user (generally the site administrator) must configure and run the CollabNet TeamForge application server in VMware Player. When the CollabNet TeamForge 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 download may also run on some other VMware products, such as VMware Workstation 5.5. However, these instructions are only for using VMware Player.

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.

What is the look project?

The `look` project contains special files that can override your site's default appearance and content, such as the default icons, fonts, colors, and labels.

Unlike most projects, the `look` project has no members. It is only visible to users with site administration permission. Its only purpose is to control your site's look and feel, including such things as fonts, background colors, icons, and the wording of the onscreen labels that appear throughout your site.

Any project on your CollabNet TeamForge site can have one or more Subversion repositories associated with it. The `look` project has just one Subversion repository. That repository is named `branding`.

When a user requests a page from your site, CollabNet TeamForge checks the `branding` repository to see if any files there specify custom fonts, colors or text strings. If such specifications are found, CollabNet TeamForge displays the page according to those specifications. If not, the page displays according to the default design.

Having your custom look-and-feel specifications in a Subversion repository enables you to roll back changes, track contributions, and use all the other features of a source code versioning system.

What wiki engine does TeamForge use?

TeamForge currently uses the JSPWiki engine to render the wiki component pages in TeamForge. You can visit the JSPWiki homepage at the link below. Please keep in mind that TeamForge does restrict JSPWiki and not all functionality found on that site applies to the TeamForge wiki (plugins, for example).

Does CollabNet TeamForge support merge tracking?

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

Any Subversion 1.5 servers you have integrated with CollabNet TeamForge 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, linked below.

What is a private IP address and what are the private IP ranges?

Any IP address that falls specified ranges is a private IP address and is non-routable on the Internet.

These addresses are reserved for use only within private/corporate network and cannot be seen outside the private networks. These private addresses are translated at the company's firewall into an external (public) IP address, which will be some address that does 'not' fall within these ranges.

10.0.0.0/8=10.0.0.0 – 10.255.255.255
192.168.0.0/16=192.168.0.0 – 192.168.255.255
172.16.0.0/12=172.16.0.0 – 172.31.255.255

An address is Private if it starts with:

10 or
192.168 or
172.[16-31]

It is possible for anyone to see their external (public) IP by going to any one of a number of sites that provide this information as a free service. One example that's easy to remember is http://whatismyip.com.

What is the vamessages.log used for?

TeamForge records just about everything it's doing to this file. You can use this log to debug startup issues, performance issues, system errors, exid backtraces, JVM issues, SQL issues, etc.

The log can be found at `/opt/collabnet/teamforge/log/vamessages.log`. Since this log contains so many different types of log messages, it grows extremely rapidly, so the file is automatically rotated by TeamForge when it reaches 100M, and TeamForge will keep the previous ten copies of the log. If you are having any kind of issue, this is probably the log to look at.

How do I use the TeamForge updater to manage backups of old versions of TeamForge?

You can safely delete the items in your <sourceforge_base_install_directory>/sourceforge_home/backups as long as you are comfortable with your version of TeamForge , and have no desire to go back. This directory may also be safely omitted from your backup plan.

How does TeamForge deliver activity reports?

The data in your reports comes from a special database that extracts live site data from the production database at intervals you specify. This protects the production database from excess reporting traffic, while keeping report content fresh.

You can specify the time at which the reporting data is refreshed from the production database. By default, the extraction takes place daily at 2:30 a.m. in the TeamForge application server's time zone.

The reporting database can be deployed on a separate machine to help channel load away from the application server. Historical data is available even if the application server no longer stores it.

Where does the reporting data come from?

The ETL app extracts its data from the live production PostgreSQL or Oracle database where the TeamForge site stores most of its critical data. (Information about reporting configurations is also stored in the production database.) Some data is also gathered from the file system.

How is the production data converted into reporting data?

TeamForge extracts a snapshot of the production data, transforms it into a format that supports reporting requirements, and loads it into the datamart, which is optimized for fast retrieval. The Extract-Transform-Load (ETL) application is a Tomcat JVM running as a TeamForge service under the TeamForge integration server architecture.

Where is the reporting data kept?

After the ETL app collects and processes the live site data, it is stored in a separate database called the datamart. If the TeamForge site uses a PostgreSQL database, then the datamart is also a PostgreSQL database; likewise for Oracle. The datamart uses a Star Schema-based design for tables.

How are the reports shown in the TeamForge user interface?

The reports are rendered in the TeamForge UI using Adobe Flex.

☞ **Note:** When a site is upgraded, there will be a delay before reporting data is available to users, until the scheduled ETL run has occurred. Performing a manual ETL run immediately after an upgrade is not advisable, since it could consume a lot of system resources leading to performance problems.

What is an integrated application?

An integrated application is a stand-alone application that can seamlessly integrate into any CollabNet TeamForge project.

You can use integrated applications to incorporate these types of applications into your TeamForge project:

- Third party applications
- Internally developed applications
- Integrations developed using the TeamForge SOAP APIs
- External websites

When you add an integrated application to your project, an icon is added to your project navigation bar. Clicking the icon displays the integrated application in the main TeamForge project window.

TeamForge site-administrators can register site-wide integrated applications that project administrators can opt to use across projects.

Site administrators or users with site-wide roles with the administration permissions for integrated applications can enable/disable integrated applications.

☞ **Tip:** Disabling an integrated application restricts it from being added to projects. However, disabling an integrated application does not affect the projects where the integrated application might already be in use.

After your site administrator registers an integrated application on the site level, on adding it to your project, an icon is added to your project navigation bar. Clicking the icon displays the integrated application in the main CollabNet TeamForge project window.

☞ **Note:** You can register and integrate as many applications per project as you wish. However, because each integrated application adds an icon to the project navigation bar, creating a large number of integrated applications can cause horizontal scrolling.

Can I control user access to an integrated application?

TeamForge can integrate the permissions scheme of a separate application into the TeamForge role-based access control system.

To look at how this works, we'll use the Pebble blogging tool as an example. Pebble is an application that you can quickly integrate with TeamForge.

Pebble brings with it a set of pre-determined roles that you can assign to project users. The roles are defined in the XML application configuration file.

Blog Reader	You can only read blogs and make comments, the comments are sent for moderation.
Blog contributor	You can add blog posts, but they will be sent for moderation.
Blog publisher	You can add blog posts, moderate comments and blog posts.

Blog owner   You can do all that a Blog publisher does as well as change the blog properties and security options.

Any site user with one or more of these roles can see the **Pebble Blog** button in their project toolbar. Clicking that button allows them to operate Pebble according to their access rights.

👉 **Note:** Site Administrators don't need any specific permissions; they have all permissions on all projects on the site.

How does an integrated application interact with other TeamForge tools?

When you integrate an external application into your TeamForge site, the application can take full advantage of object IDs, links and Go URLs.

To look at how this works, we'll use the Pebble application as an example. Pebble is a blogging tool that you can quickly integrate with TeamForge.

Prefix

Object IDs uniquely identify a TeamForge object so that you can access and use it in different contexts. For example, to get to artifact `artf1234` quickly, you just enter `artf1234` in the Jump To ID box. In the Pebble tutorial application, the date of a blog post, in YYYYMMDD format, is used as the object ID.

A prefix is an alphanumeric string attached to the beginning of an object ID that TeamForge uses to manage object IDs from different tools. For example, in the Pebble app, `<prefix>_20100601` gets you a page showing all the blog posts in the project that were published on June 1, 2010.

The prefix can either be the one specified when an integrated application is added to a project by project administrator, or the one in the XML Application configuration file depending on the "require-per-project-prefix" setting. The "require-per-project-prefix" setting can be true or false. If it is false, each project integration would not need to provide a project prefix; so the one provided in the XML Application configuration file takes effect. If the "require-per-project-prefix" setting is true, a prefix needs to be provided by the user during every project association.

The amount of information the prefix carries depends on the kind of application you are integrating into your TeamForge site.

- With applications that use object IDs, such as Project Tracker and JIRA, you can identify the project that the object belongs to from its object ID.
- For applications that don't have uniquely identified objects, or don't have the notion of "project," such as MoinMoin or Review Board, you can choose a prefix that's specific to the project where the integrated tool is used.

Go URLs

Go URLs allow a user to get to a particular object ID with a short, handy URL. To use this for Pebble, construct a URL like this: `https://mysite.com/sf/go/<prefix>_<date in format YYYYMMDD>`.

For example, if the Pebble tool in your project has the prefix `PA`, and you want to send someone all the blog posts published on app June 1, 2010, send them this link: `https://mysite.com/sf/go/PA_20100601`.

Associations

The object ID can be used to associate objects with other TeamForge objects. For example, if you want to associate a document with the blogs published on June 1, 2010, go to the document's **Associations** tab and add an association to `PA_20100601` as the object ID.

Automatic links

When you type text of the format `<prefix> _<date in YYYYMMDDD>` in any TeamForge text field, the text is converted to a link. When you click the link you see the blog posts for that date, if any.

Common errors in TeamForge

Common problems and solutions in TeamForge.

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

CollabNet TeamForge 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 application server is not running.
- Your organization has exceeded your maximum number of licensed users for the CollabNet TeamForge 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 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 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 VMware image to drift from that of the host machine. If you notice this issue, you can set the CollabNet TeamForge 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*

Why does my CollabNet SourceForge Enterprise 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 SourceForge Enterprise VMware image to drift from that of the host machine. If you notice this issue, you can set the CollabNet SourceForge Enterprise 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*

Why won't my CollabNet SourceForge Enterprise virtual machine installation start?

CollabNet SourceForge Enterprise 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 SourceForge Enterprise application server is not running.
* Your organization has exceeded your maximum number of licensed users for the CollabNet SourceForge Enterprise 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 SourceForge Enterprise 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 don't the branding repo changes get rendered into UI?

It may be due to the property 'subversion_branding.repository_base' pointing to /sf-svnroot instead of the /svnroot directory, which is used by the scm-integration of the csfe installation.

First, check the location of the branding repository in subversion_branding.repository_base=/sf-svnroot' in `/opt/collabnet/teamforge/runtim/conf/sourceforge.properties`.

If it has to be /svnroot, then add an entry that states SUBVERSION_BRANDING_REPOSITORY_BASE=/svnroot

Then re-create a runtime and restart TeamForge.

How can I solve the PSQLException when starting the app server after changing my DB server IP address?

You might need to replace the old IP address with the new IP address in the `<connection-url>` block of the file sourceforge-postgres-ds.xml, located in
`/opt/collabnet/teamforge/jboss/jboss-3.2.6/server/default/deploy`.

This issue may occur after changing the IP address of the database server and making the corresponding IP changes in pg_hba.conf, /etc/hosts and postgresql.conf files. The following error may appear when starting the application server though you are able to login to the DB server from the app server.

```
[JBossManagedConnectionPool] Throwable while attempting to get a new connection:
 null
org.jboss.resource.JBossResourceException: Could not create connection; - nested
throwable: (org.postgresql.util.PSQLException: Connection refused.
Check that the hostname and port are correct and that the postmaster is accepting
 TCP/IP connections.
```

Why do search and email server show "Could not connect"?

Typically this means the tomcat container for James and the search service are not running. You can restart this with the commands shown below. You may need to set the JAVA_HOME environment variable to the location of your JDK.

```
sh /opt/collabnet/teamforge/dist/james/james-2.2.0/bin/phoenix.sh stop
sh /opt/collabnet/teamforge/dist/james/james-2.2.0/bin/phoenix.sh start
```

Why is the password and login shell changed for users on my cvs/svn server?

TeamForge uses local user accounts on the SCM server to provide access to CVS repositories via ssh. If any local user accounts on the SCM server match user names within TeamForge they will be changed.

If you are planning to use CVS in the SCM server, you should ensure that the local accounts do not conflict with the TeamForge user accounts. Adding a prefix to the local user accounts (local_username) would be one way to resolve this and prevent the usernames from conflicting. Alternatively, if you are not using CVS repositories, the CVS integration can be removed altogether.

JBoss crashed with out of memory error, how do I prevent this?

This can indicate that the JVM heap size is set too small. You can adjust this by changing the -Xms and -Xmx settings of the JBOSS_JAVA_OPTS token in site-options.conf and rebuilding runtime.

This will appear if the JBoss application server has crashed and you find this error in the server.log:

```
INFO [STDOUT] java.lang.OutOfMemoryError: Java heap space
```

The default maximum heap size of 640MB can cause issues on a heavily used site. If the CTF application is the only thing running on the server, you can increase this to half of the total physical ram on the machine. This should still allow enough memory for the OS and other necessary processes. If you are also running the app, database and scm on the same machine a maximum heap size of 1/4 or the total ram maybe a better setting. Determining the right JVM settings for your install will require testing with your particular usage patterns and database.

You can view the current memory usage under the JVM Environment section of the JBoss webconsole at `http://<CTF_SERVER>:8080/web-console/`. You will need to log in using the CTF admin password.

Why do I get a JBoss error - "failed to start in 240 seconds, giving up now" - while installing TeamForge?

You get this error when the system's RAM is less than the minimum recommended value of 4GB. However, it's most likely that JBoss will start within a few minutes.

To make sure that JBoss starts up, check the `service.log` file using this command:

```
    tail -f /opt/collabnet/teamforge/log/apps/service.log
```

If you see messages like the following, the TeamForge application will start in a few minutes.

```
        Check Port Available PASSED: Port 4444 on localhost is available
        Check Port Available PASSED: Port 4445 on localhost is available
        Waiting for application server to start up.. this can take a few
minutes.
```

Why don't help links in TeamForge work after upgrade from SourceForge Enterprise 4.4?

After upgrading you may need to force the help links to point to the remote source. This can be done by uncommenting the following lines from site-options.conf and rebuilding runtime. HELP_AVAILABILITY=remote REMOTE_HELP_URL=http://help.collab.net

Why am I getting "Could not connect" status for my email and search server?

On the System Tools page, when you see "Could not connect status for search and email servers," you must stop and start your phoenix.sh process.

You may also need to set the *JAVA_HOME* environment variable to the location of your JDK.

The stop/start Phoenix commands:

```
sh /opt/collabnet/runtime/scripts/phoenix.sh stop
sh /opt/collabnet/runtime/scripts/phoenix.sh start
```

Why is my email taking a long time to arrive?

TeamForge uses the James MTA to send and parse all email coming to and from the system. In this case, the best course of action is to look in the james mailet logfile.

Your logfile will help you to determine what is going on with the emails that are being sent from your system. Your logfiles will look very similar to this:

```
07/02/07 07:54:43
INFO James.Mailet: ?RemoteDelivery:
Attempting delivery of Mail1170135534355-39088-to-domain.invalid to host
domain.invalid at 192.168.0.1 to
addresses [invalid.user@domain.invalid] 07/02/07 07:55:43
INFO James.Mailet: ?RemoteDelivery: Could not connect to SMTP host: 192.168.0.1,
 port: 25; nested exception
is: java.net.?ConnectException: connection to 192.168.0.1 timed out 07/02/07
07:58:43
INFO James.Mailet: ?RemoteDelivery: Storing message
Mail1170135534355-39088-to-domain.invalid into outgoing after 7 retries 07/02/07
07:58:43 INFO James.Mailet: ?RemoteDelivery: Attempting delivery of
Mail1170831482124-2756-to-company.com to host mx.company.com. at 127.0.0.1 to
addresses
[good.user@company.com]
```

As you can see, the James MTA stores outgoing emails to resend at a later time. These files can be located in this directory: /opt/collabnet/teamforge/james/james-<ver>/apps/james/var/mail/outgoing/

When you do a directory listing of the files, you will see a listing of files very similar to this:

```
4D61696C313137303833332313134303131322.Repository.?FileObjectStore
4D61696C313137303833332313134303131322.Repository.?FileStreamStore
```

The FileObjectStore is a binary file, but, the FileStreamStore can be viewed with an editor or your favorite paging program in order to determine the contents. Sometimes, the directory can grow to a large number, where you will not be able to use a standard bash expander to delete all of the files. In that case, use the following shell script:

```
for i in * ; do /bin/rm $i; done
```

to remove all of the objects from the outgoing directory.

Due to firewall restrictions I cannot send email from James. How can I resolve this?

If James is unable to send email directly due to firewall restrictions, or mail being rejected from the application servers IP address, you may have to configure it to use a gateway mail server to send outgoing messages through.

To do this, you will need to add the following to the `<mailet match="All" class="?RemoteDelivery">` directive in the james config file at
`/opt/collabnet/teamforge/james/james-<version>/apps/james/SAR-INF/config.xml`:

```
<gateway>smtp.example.com</gateway>
<gatewayPort>25</gatewayPort>
```

You should find these commented out on line 362 of the config file. If your gateway mail server requires authentication to send email, you may also add the following directives:

```
<username>username</username>
<password>password</password>
```

Why do we have errors creating or altering repositories and adding or removing users?

The TeamForge SCM Integration server runs an instance of Tomcat and then launches TeamForge inside the Tomcat container. If you are experiencing issues creating or altering repositories or adding and removing users from repository access, and the other TeamForge integration logs are not providing any clues, you may wish to review the Tomcat log at: `/opt/collabnet/teamforge/tomcat/jakarta-tomcat-<version>/logs/catalina.out`.

Sometimes, OS-level errors will be flagged into this log and not others. In our experience, it is pretty rare to find something in this log that is not logged elsewhere.

Why does the SOAP service show "could not connect" on the Server Status page when everything else appears to work?

This can be caused by an incorrect host name in /etc/sourceforge.properties. Rebuilding runtime will correct this, assuming the hostname is set correctly in the site-options.conf file.

This issue can occur when using the restore.py script to restore data from a TeamForge instance with a different hostname.

Why do I get a server status error when I perform a search?

Occasionally, an exceedingly large or complex document causes the search indexing service to abort. This is typically when all searches in TeamForge return an exid to the user.

Check the server status page and see if the search server is listed as anything other than OK. If it is not OK, then you should restart the search service by logging into the TeamForge application server as root and issuing the following commands:

```
/opt/collabnet/teamforge/james/james-/bin/phoenix.sh restart
```

Check the server status page again in TeamForge and ensure that it shows a status of OK. If it shows OK, then searches should now work, and the site will slowly catch up on any indexing requests that were logged while the service was down. If you continue to get exids returned for all searches even with an OK status, then you probably have corrupt search index files and you should see the link below.

Why would some users not get email?

Check your dnsserver-<date and time>.log.

James records all errors related to resolving DNS for outbound mail to:
`/opt/collabnet/teamforge/james/james-<version>/apps/james/logs/dnsserver-.log`. If you find that some of your TeamForge users are receiving email, but significant groups of others are not, you should consult this log to determine if James is experiencing difficulties in resolving their domain or MX records.

What does the "psql: could not connect to the server: No such file or directory" error message mean?

This error indicates that the PostgreSQL database server is not running. You need to restart the server.

Use the following command to start the server:

```
service postgresql start
```

Why do I get a proxy timeout when I try to view certain SCM pages?

If you are getting a proxy timeout error when you try to view a SCM page, you may need to configure the *Apache 2.2 Proxy Timeout* to 300 or less in the *httpd.conf* file.

If you get the following error while attempting to view a SCM page in SFEE:

```
The proxy server received an invalid response from an upstream server.
The proxy server could not handle the request
    GET
/integration/viewcvs/viewcvs.cgi/ibe-rules/tags/phases/ibe-rules_09.02.0-Ph-200902_test_20090105/
Reason: Error reading from remote server
```

Configure your *Apache 2.2 Proxy Timeout* to 300 or less in the *httpd.conf* file.

When using SourceForge Explorer version 4.4 SP1 to connect to SourceForge 4.4 SP1, I receive the following error message: "Error - wrong number of arguments".

This error is due to an incomplete configuration of the SourceForge server. To correct this, add the following to `/opt/collabnet/teamforge/sourceforge_home/etc/sourceforge_configuration.properties`:

```
sf.requireAssociationOnDocumentCreate=true
    sf.allowedAssociationTypeOnDocumentCreate=Any
    sf.requireArtifactToBeOpenOnDocumentAssociation=false
```

After restarting the application you should be able to connect using SourceForge Explorer.

Why do I get a URL "not found" or "moved permanently" error after applying a patch/upgrade?

If you are experiencing a URL "NOT FOUND" or "MOVED PERMANENTLY" error after applying a patch or upgrade Apache ProxyPreserveHost token to on in the httpd.conf file.

If you have applied a patch or upgrade and are now receiving the following error:

```
<The document has moved <a
href="https://www.parkingforge.com/sf/global/jsp/buildtime.html"
    format="html" scope="external">here</a>.</p>
<hr> <address>Apache/2.2.3 (Red Hat) Server at www.parkingforge.com Port
80</address> </body></html>
Not Found


The requested URL
/sf/sfmain/do/userPicker/projects.pftool//sfmain/do/listMonitoringUsers/projects.pftool/discussion.announcements
 was not found on this server
```

Or if you are trying to add users to a monitoring list, and are receiving the following error:

```
Not Found
The requested URL

/sf/sfmain/do/userPicker/projects.pftool//sfmain/do/listMonitoringUsers/projects.pftool/discussion.announcements

    was not found on this server.
```

Set the `ProxyPreserveHost` token to ON in the `httpd.conf` file.

Why can't TeamForge send my outbound mail?

If you are unable to send email directly due to *firewall* restrictions, or if mail is being rejected by the application server's IP address, configure TeamForgeto send outgoing messages through a gateway mail server.

Configure TeamForge to send outgoing message through a gateway mail server by adding the following to the `<mailet match="All" class="RemoteDelivery">` directive in the configuration file at `/opt/collabnet/teamforge/`*user*`/`*user*`-<version>/apps/`*user*`/SAR-INF/config.xml`:

```
<gateway>smtp.example.com</gateway>
<gatewayPort>25</gatewayPort>
```

☞ **Note:** The code is usually found on line 362 (commented out) of the configuration file.

If your gateway mail server requires authentication to send email, you may also add the following directives:

```
<username>username</username>
<password>password</password>
```

Why do I get a TeamForge system error in the project template creation page?

This may be because of a few stale permissions in the project in which you are trying to create the template. You can resolve this by identifying and deleting the stale records using this SQL.

```
select role_id from role_operation ro left outer join ia_project_association ia

on (ro.resource_value = ia.id) where ia.id is null and resource_value like
'%prpl%' ;
```

```
select role_id from operation_cluster ro left outer join ia_project_association
 ia
on (ro.resource_value = ia.id) where ia.id is null and resource_value like
'%prpl%' ;
```

TeamForge roles and permissions

Questions about setting permissions and using roles in TeamForge.

Can I set permissions so that users can move documents but not delete them?

You cannot configure document management permissions so that a user can move documents but not delete them.

It is not possible to separate move and delete permissions, because a move is actually a copy/delete action. The document is not really moved, it is copied to the new location and then deleted from the original location.

Can I grant users permissions to create their own accounts?

Yes, as of SourceForge 4.1 SP3, it is possible to configure TeamForge such that new users can create their own accounts, but the accounts are not activated until a Site Admin approves them.

To enable this mode of operation, simply add the following line to /opt/collabnet/teamforge/sourceforge_home/etc/sourceforge_configuration.properties: `sf.approveNewUserAccounts=true`. Once this line has been added to the file, restart TeamForge for it to take effect.

The Site Admin can still create accounts for new users and they will not be held for approval. Also note that the user will receive an email from TeamForge telling them to confirm their password by clicking on the given link, and the link will not work. The password is properly set on account approval.

Why can't Oracle connect to my TeamForge installation?

The simplest way to correct this is to overwrite the.jar included with TeamForge with the one from $ORACLE_HOME.

TeamForge uses the thick Oracle JDBC driver, which has two parts. One of these is provided by TeamForge, the other is in $ORACLE_HOME. If these two components are incompatible, TeamForge will be unable to make a connection to the database.

Follow these steps to overwrite the.jar included with TeamForge with the one from $ORACLE_HOME:

```
cp $ORACLE_HOME/jdbc/lib/ojdbc14.jar
        /opt/collabnet/teamforge/jboss/jboss-3.2.6/server/default/lib/
```

A restart of the application will be required to use the new.jar.

Why does SearchReindex.sh fail on SourceForge 4.4?

This occurs as the search indexer included with SFEE 4.4, requires the 4.3 java SDK. You can obtain the java 4.3 SDK at the link below. Extract the file to `/opt/collabnet/teamforge/sourceforge_home/sfsoap` (assuming you installed SourceForge into `/opt/collabnet/teamforge`).

Running the SearchReindex.sh script on a clean SFEE 4.4 install sometimes fails with the following message: Soap43 SDK must exist: `/opt/collabnet/teamforge/sourceforge_home/sfsoap/sf_43_sdk/soap/lib`.

After installing the 4.3 SDK, you may receive the following error:

```
Could not login (; nested exception is:
  java.lang.NullPointerException)
```

This is due to an older version of axis.jar being included in the 43 SDK. You can correct this by copying a newer version of axis.jar into the sf_43_sdk directory:

```
cp /opt/collabnet/teamforge/sourceforge_home/lib/axis.jar
   /opt/collabnet/teamforge/sourceforge_home/sfsoap/sf_43_sdk/soap/lib/
```

You may need to substitute `/opt/collabnet/teamforge` for the location of your SourceForge install.

Tasks in TeamForge

Questions about performing specific tasks in TeamForge.

What happens when log files get too big?

Log files can grow very large over time. To maintain reasonable log file sizes, you can rotate logs on a schedule.

When you rotate logs automatically, live logs are archived every day at 00:00.

Archived logs are stored in compressed form in a directory alongside the live log. For example, if live logs are stored at `<LOG_DIR>/{apps, apache,...}`, then compressed log archives are stored at `<LOG_ARCHIVE_DIR>/{apps, apache,...}`.

The directory structure of the log directory is preserved in the log archive directory.

☞ **Note:** Empty log files are not compressed.

What is the suggested log configuration for a production system?

To troubleshoot installation issues, the default log4j configuration is set to DEBUG. This can cause the log files to become quite large. Once your system is successfully installed and in use, you should drop the log levels down to INFO.

See *Change the logging level on your site* on page 92 for how to do this.

If you still have a problem with over-large log files, you may want to set up log rotation. See *Rotate log files*.

How do I enable post-commit logging?

You do this by editing the post-commit.py file.

Edit the `/opt/collabnet/teamforge/runtime/sourceforge_home/integration/post-commit.py` file.

Search for log.setLogging(False) and modify the value from False to True.

How do I make the monitoring messages be sent from Forge Administrator?

You can change the default behavior for site options by changing the value from "false" to "true" in this statement: # MONITORING_EMAIL_FROM_ADMINISTRATOR=false

If the site option MONITORING_EMAIL_FROM_ADMINISTRATOR=true, then "From:" field is the Forge Administrator, else it is from the user who made the change that initialized the monitoring email.

How can I remove the RHEL test page after TeamForge installation?

You can modify the httpd.conf file to remove the RHEL test page that appears in place of the home page of TeamForge after installation.

Add COLLABNET CONFIGURATION to the /etc/httpd/conf/httpd.conf file. This includes the rewrite rules, which removes the RHEL test page.

How can I find the number of files in a repository without checking it out?

To list the files in a repository use the command: `svn ls -R file:///svnroot/REPONAME| wc -l` (Requires local access to server)

How do I connect to the Teamforge Postgres database?

For onsite installations, use the commands below.

Note these commands will only work for onsite installations - direct server access is required.

```
ssh to Teamforge server
```

```
    sudo su - postgres
    psql -d ctfdb
```

How do I generate a wiki table of contents?

You can create a table of contents from any heading text that you have in your wiki page.

For versions 5.2 and earlier, generating a wiki table of contents requires the Wiki TOC plugin, available through CollabNet Professional Services.

To enable TOC for a Wiki page, place the following in your Wiki page at the spot where you want the Table of Contents to appear.

```
%%insert-toc
%%
```

The Table of Contents is generated automatically based on the heading markers in the wiki page, e.g. `!!!Heading`

What is the correct procedure for modifying a hosted Lab Manager profile?

All profile modifications must be done through the Lab Management UI, under Administration > Manage Profiles. Lab Manager profiles should not be directly modified and changes should not be committed to subversion.

To modify your profile, follow these steps:

1. In the browser, login to https://mgr.cubit.domain.com/
2. Click on Administration
3. In the left pane, click on Manage Profiles
4. Click on the profile (your_profile_name)
5. Click on the Packages tab and choose your options

Can I add custom links to the Quick Links section of the application home page?

Yes, as of SourceForge 4.3 Hotfix 5, it is possible to add your own custom links to the Quick Links section of the application home page by editing file sourceforge_configuration.properties.

To do so, simply open the file `/opt/collabnet/teamforge/sourceforge_home/etc/sourceforge_configuration.properties` and add the following:

```
quicklink.link0.text=Your Descriptive Text Here
quicklink.link0.url=http://www.yourcorp.com
quicklink.link0.description=Detailed but succinct description of the link
```

After saving the file and restarting TeamForge, your links will become active. To add additional links, simply increment the link0, e.g. link1, link2.

How do I configure the timeout for Apache in TeamForge?

This can be changed by editing the setting WWW_SERVER_TIMEOUT.

The timeout for Apache (httpd) can be configured in seconds. Default is 300 seconds. You must re-create the runtime for the change to take effect.

How do I back up TeamForge?

TeamForge has essentially four data components that require the System Admininstrator to proactively back up in case of system failure. These components are:

- the database
- the filestorage
- the search indexes

- the scm data

As with most large and complex applications, the recommended method of backing up TeamForge involves shutting down the application. While it is possible to back up the application while it is up, the backup itself will not provide 100 percent data consistency. You should undertake one of these live backups only as a last resort and only after ensuring all the potential consequences are fully understood.

If, for whatever reason, a full, offline backup is not possible, you should do the following to ensure that as little data is being changed as possible. While these steps won't completely replicate the offline backup, they can mitigate most of the issues w/ doing a live backup. Again, we do NOT recommend this. Offline backups are your friend. Just schedule the downtime and take the hit.

For the CVS repositories, you should use whatever normal filesystem backup method your company prefers, much like the FILESTORAGE section above. For SVN repos, it is highly advised that you use the 'svn dump' action to export them to normal files and then back up those dump files.

Please note that due the nature of SFEE, there are several interdependencies between these various data storage points. As such, you must take great care to ensure that all these components are backed up AT THE SAME TIME. You will NOT be able to use a database backed up yesterday morning with a filestorage from the night before.

Quieting the system

First, you'll want to turn off the web server so users can't get to the UI (assuming you are routing through Apache and not direct to JBOSS): `service httpd stop`

Second, on the SCM server, remove the execute bits on the SCM: `chmod -x /usr/bin/{cvs,svn}`. This will prevent anyone from executing 'cvs' or 'svn' for new checkins/checkouts/tags. Please note however, that it won't affect existing running processes. Either wait for them to finish, or kill them.

Finally, pause the search indexing engine: `touch /opt/collabnet/teamforge/var/searchIndexes/LOCK_INDEXES`

The database

The majority of TeamForge data is stored in the database. To back up the database, follow the recommended procedure from your db vendor. If you're using PostgreSQL, you can use this command: `pg_dump -Fc ctfdb > ctfdb.dump`

The file storage and search indices

Any documents uploaded to TeamForge, or attachments to an artifact or forum, as well as the search indices, are stored on the filesystem as normal system files. Please use whatever normal backup method you use to ensure filesystem restorability (tar, dd, cpio, Ghost, etc). You should back up the following directory and ALL its subdirectories: `/opt/collabnet/teamforge/var`

The SCM data

On the SCM box, you will also need to back up your SCM data. This data is contained in your companies various repositories which are located under: `/cvsroot` and `/svnroot`

Finishing

Once the backup is complete, remove the file: `/opt/collabnet/teamforge/var/searchIndexes/LOCK_INDEXES`

Turn on the execute permissions for the SCM binaries: `chmod +x /usr/bin/{cvs,svn}`

And restart Apache: `service httpd restart`

How do I move an existing CVS repository into TeamForge?

Use the steps below to import and manage an existing CVS repository with TeamForge.

1. Stop CVS access to the old repo

```
chmod -x /usr/bin/cvs
```

2. Tar the old repo

```
cd /cvsroot/old_repo
tar zcvf /tmp/old_repo.tar.gz
cd..
mv old_repo /tmp
```

3. Restore CVS access

```
chmod +x /usr/bin/cvs
```

4. Transfer the repo to the TeamForge CVS server

5. Create the new repo from within TeamForge

```
Browse to your project
Click the Source Code button
Create your new repo
```

6. Untar the old repo

```
cd /cvsroot/new_repo
tar zxvf /tmp/old_repo.tar.gz
```

7. Synchronize permissions

- Login as a TeamForge site admin
- Click the Admin link
- Click the Integrations button
- Check the CVS integration you want
- Click the Synchronize Permissions button

8. Verify the new repo

9. Remove the old repo

```
/bin/rm -r /tmp/old_repo
```

How do I move an existing SVN repository into TeamForge?

If you have an existing SVN repo that you would now like to manage with TeamForge, follow the steps below.

1. Stop SVN access to the old repo

2. Dump the old repo

```
svnadmin dump /svnroot/old_repo > /tmp/old_repo.dmp o mv /svnroot/old_repo
/tmp
```

3. Restore SVN access

4. Transfer the repo to the TeamForge SVN server

5. Create the new repo from within TeamForge

6. Browse to your project and click the Source Code button, then create your new repo

7. Load the old repo

```
cat /tmp/old_repo.dmp|svnadmin load /svnroot new_repo
```

8. Synchronize permissions

- Login as an TeamForge site admin
- Click the Admin link
- Click the Integrations button
- Select the SVN integration you want
- Click the Synchronize Permissions button

9. Verify the new repo

10. Remove the old repo

```
/bin/rm -r /tmp/old_rep
```

Where do I configure my client proxy settings?

Configure proxy settings in the servers file (created when Subversion is installed on your system).

The server file is created when you install TortoiseSVN, Eclipse or command-line Subversion. Use the appropriate path from the installation folder to configure proxy settings:

CLI Unix/Linux	/home/<username>/.subversion/servers
CLI Windows	C:\Documents and Settings\<username>\Application Data\Subversion\servers
Eclipse	Window > Preferences > General > Network Connections
IE (6/7)	Tools > Options > Advanced Network > Connection Settings
Firefox	Tools > Internet Options > Connections tab > LAN Settings
TortoiseSVN	Windows Explorer > File > TortoiseSVN > Settings > Network (by default, TSVN uses browser proxy settings)

How do I make TeamForge work the same when the IP address of the server changes?

Update the /etc/hosts file on the server and the dns record(s) with the new IP address.

When the IP address of the server changes, update the /etc/hosts file on the server and the dns record(s) with the new IP address.

In addition, for TeamForge instance using PostgreSQL database, the IP address used by the PostgreSQL database needs to be changed in the following files:

* /var/lib/pgsql/9.0/data/pg_hba.conf - Change the IP of the host entry pointing to the SouceForge server.
* /var/lib/pgsql/9.0/data/postgresql.conf - Change the application server's IP for the listen_addresses variable. Restart Postgres service for the changes to take effect.

How do I capture the output of "top" command?

Top is the realtime monitor of the running processes in a Linux system. To log the top running processes, use the following command: `top -b -n 1`.

-b = Batch mode operation - Starts top in 'Batch mode', which could be useful for sending output from top to other programs or to a file.

-n = Number of iterations limit as: -n number specifies the maximum number of iterations, or frames, top should produce before ending.

CollabNet TeamForge 6.1.1 release notes

Last Revision: November 2011

Look here for information about changes in CollabNet software, such as new features, fixes and late-breaking updates.

New features in CollabNet TeamForge

CollabNet TeamForge 6.1.1 adds these new features.

Core infrastructure	TeamForge 6.1.1 supports these upgrades:

- JBoss 6.0_26
- Tomcat 7.0.22
- Postgres 9.0

Section 508 compliance

- There's better input support in forms using assistive technologies
- Text equivalents are available for images
- For information conveyed with color-only, there are alternatives without color
- Row and column headers are available for assistive technologies
- Links are available for applets and plugins

Installer improvements	The YUM based installer improves installing, upgrading and patching TeamForge.

Search

- TeamForge 6.1.1 offers better global search and search specific to a project tool.
- The Lucene library has been upgraded to the latest release 3.2.0.

Integrated applications

- General Availability (GA) release of the Integrated Application Framework (IAF)
- Integrated applications offer a detailed view displaying deployment and application properties and project parameters. You can export existing definitions, edit them, and make content from third party tools available in project pages.

Site level metrics	Site administrators can *monitor source code commits* across all projects at the site level.
Data entry validation mechanisms	In TeamForge 6.1.1, the SOAP_ARTIFACT_LIST_LIMIT token (in site-options.conf) can be used to limit the number of artifacts returned via SOAP calls. In TeamForge releases earlier than 6.1.1, SOAP calls returned everything that was asked for, and that is the default behavior in TeamForge 6.1.1 as well (with this token set to -1). However, sites with performance and stability issues (OutOfMemory errors) in returning a large number of artifacts can now limit the number using this token. Changing this value requires a recreate-runtime and thus a site restart. Increasing the number of retrieved artifacts beyond the optimal 20,000 - 25,000 range might cause a heap dump. See *this wiki page* for related information.

Reports

- Quartz has been upgraded to version 1.8.3.
- Wiki and Docman are converted to Enterprise JavaBean 3 (EJB 3) classes.

IAF	Debug information has been introduced for integrated applications.

Issues resolved in CollabNet TeamForge

CollabNet TeamForge 6.1.1 resolves these issues.

Wiki

ID	Description
artf78730	The Preview Changes button was not available when a wiki page was edited using the WYSIWYG editor.
artf83648	Having a large number of links on a wiki page affected its performance. Wiki performance has now been improved.
artf96788	On clicking Show Details, wiki pages sometimes displayed "null" instead of version information in the Safari browser.
artf96874	A version comment added while editing a wiki page was retained for subsequent updates.
artf99925	On wiki pages, not all object IDs were getting converted to links.

Role based access control

ID	Description
artf96276	In the Roles tab, component IDs are now shown as links.

Integrated application

ID	Description
artf63826	When making a SCM commit that involved an integrated application, the Association tab for the commit object was not showing the integrated application's ID.
artf100146	Integrated application variables can now be edited even after the application is integrated.
artf102389	The pending count was not displayed on the Pending SCM Integrations and Access Request tabs for existing SCM requests.

SCM

ID	Description
artf104350	Publishing repositories were getting corrupted because they did not follow the three-directory structure (bdb_data, bdb_log and bdb_env). This has been fixed by using `convert_publishing_repository_to_bdb_structure.py` to create publishing repositories in a three-directory structure.
artf103190	An error occurred when you tried to add Subversion Edge as an integration server.

Tracker

ID	Description
artf96598	Under rare circumstances, assigning an artifact to a planning folder resulted in a system error.

Remote Publishing

ID	Description
artf63225	Pages created in Microsoft Word or Frontpage did not render correctly.

Tracker Admin

ID	Description
artf63820	It was not possible to delete tracker fields that ended with a backslash.

My Workspace

ID	Description
artf91530	The Organization Name was eliminated for all expired user accounts.

Discussions

ID	Description
artf96477	Reply hierarchy was not handled correctly when posting replies via email to a topic thread in the discussion forum.
artf98213	Users were not able to receive mail sent to user@domain.

Soap

ID	Description
artf102989	A project group could not be removed from a project through the API.

Soap

ID	Description
artf76723	It was not possible to configure the path-based access control of a Subversion repository with non-ASCII characters.

IAF

ID	Description
artf106228	Users with Project Tracker permission via user groups were not able to view Project Tracker.
artf98819	Integrated application information can now be viewed in the **Admin** > **Integrated Apps** page. page.

Subversion

ID	Description
artf105143	Entries were not registered in the `previous_last_login` column of the `sfuser` table when a checkout or commit was made in the Subversion repository.

Installer/Updater

ID	Description
artf100681	The default value of JBoss `MaxPermSize` has been increased to 512M.
artf72306	The maxHttpHeaderSize connector definition for Jboss and Tomcat services is configured as a site option.

Perforce

ID	Description
artf93024	It was not possible to create a repository in the Perforce integration server.
artf105694	Text files could not be committed the Perforce repository.

CEE Migration

ID	Description
artf63826	Commit objects were not getting created for the Project Tracker projects.
artf101185	After the upgrade, it took some time for the publishing repositories to get created for projects imported from other TeamForge sites.
artf98557	Migratorator and Schematorator used different thresholds for converting from varchar to text.

Reporting

ID	Description
artf100816	The Subversion Edge repositories created in TeamForge were treated as Subversion repositories in reporting.
artf91546	The ETL server status was not available in System Tools.

Known issues in CollabNet TeamForge

The following known issues in CollabNet TeamForge 6.1.1 will be addressed in upcoming patches or releases. Some of them have workarounds.

- For an Integrated Application variable, when you specify an ID in upper case, for example, an id_pattern BCA[0-9]+, there is a problem because TeamForge converts the ID to lower case.

  Workaround: Specify any case for id_pattern. For the example, this is [bB][cC][aA][0-9]+.

- Global search and discussion search display posts awaiting authorization for users with the "Discussion – View" permission.
- For MS Excel documents created using MS Office 2007 or 2010 (in `xlsx` format), wikis fail to convert the content to HTML. The error "Unable to recognize OLE stream" occurs when the content is rendered in a wiki.

  Workaround: Save MS Excel documents created using Office 2007 or Office 2010 with the file type "Excel 97 -2003 workbook" to fix the problem.

- JavaScript is not supported in news posts.
- A Project Tracker user with Assignable permission should also have the Query permission to be able to "Refresh" the list of users in a pop-up for a user attribute.

- In Project Tracker's user picker pop-ups, any available user group's name always appears as "$user.Name".
- While installing TeamForge on Oracle or Postgres, you might see a warning message such as "Connection problem occurred with Postgres...". If you see this message while recreating the runtime, you can safely ignore it.
- When you run the fix-data_permissions.sh script on a BTF site (where the SAAS_ENABLED token is set to false), a "unary operator expected" error appears.

  Workaround: In BTF mode, run the script with options: `./fix_data_permissions.sh 5 5`

  When the script runs, its output indicates that the permissions of bdb_log, bdb_env and bdb_data directories - which are not yet present in the system - are being fixed.

  Also, the script displays the message "cannot access `/svnroot/*/*': No such file or directory" in these situations:

  - On a fresh box, if the site was started before running the script
  - On a migrated box, when the svnroot directory has no repository directories

- When you install or upgrade to TeamForge 6.1.1 without replacing the HOST_localhost token (in `site-options.conf`) with the hostname, ViewVC pages for Subversion and CVS repositories created after the site is up, are rendered with a CSS error.
- Tracker search does not always return consistent results when you click the pagination link or change the number of items displayed per page.
- On all installs and upgrades, dedicated or advanced, it is mandatory to include the token "SCM_DEFAULT_SHARED_SECRET" in the `site-options.conf` file of the primary TeamForge server and give it a value of 16-24 characters for. You need to use that same key in the external SCM integration server also.