CollabNet TeamForge 7.0 Administration Guide

Contents

TeamForge system administrator how-tos.....6
Install CollabNet TeamForge 7.0.....6
Plan your CollabNet TeamForge 7.0 installation.....6
Set up networking for your TeamForge box.....7
Install CollabNet TeamForge 7.0 on Red Hat/CentOS.....8
Install CollabNet TeamForge 7.0 on SuSE.....62
Install TeamForge on VMware Player or VMware ESXi.....114
Upgrade to TeamForge 7.0.....118
Plan your upgrade to TeamForge 7.0.....118
Upgrade to TeamForge 7.0 on Red Hat/CentOS.....119
Upgrade to TeamForge 7.0 on SuSE.....173
Upgrade to CollabNet TeamForge 7.0 on a virtual machine.....221
Is my TeamForge site "dedicated" or "advanced?".....223
Enable reporting while upgrading from TeamForge 6.1.1 to 7.0.....224
Troubleshooting: Upgrade PostgreSQL manually.....224
Install a different build of TeamForge 7.0.....225
TeamForge 7.0 Patch.....226
Patch CollabNet TeamForge 7.0 on Red Hat/CentOS.....226
Patch CollabNet TeamForge 7.0 on SuSE.....228
Patch TeamForge 7.0 on VMware Player or VMware ESXi.....230
Maintain your TeamForge 7.0 site.....232
Upgrade CLI reports to the latest version on Red Hat, CentOS or SUSE.....232
Install Indexer on a separate server.....232
Supply your TeamForge license key from Teamforge user interface.....234
Supply your CollabNet TeamForge license key as a text file.....234
Support CollabNet TeamForge System Administrators.....235
Protect your CollabNet TeamForge site.....245
Get information about a CollabNet TeamForge site.....249
Rebuild runtime without the install directory.....250
Turn on site-wide reporting.....250
Synchronize TeamForge source control integrations.....251
Provide more than one source control server.....251
Upgrade Subversion on RedHat or CentOS.....252
Upgrade Subversion on SuSE.....252
Change your site's domain name.....253
Specify DNS servers.....253
Optimize PostgreSQL with vacuum.....254
Change the location of a log file.....254
Change the logging level on your site.....254
Raise the logging visibility of selected database requests.....255
Rotate TeamForge log files.....256
Schedule data extraction for reporting.....256
Removing users from monitoring objects.....257
Back up and restore CollabNet TeamForge data.....257
Move the datamart to a separate box.....259
Integrate TeamForge 7.0 with other tools.....261
Integrated application example: Pebble.....261
Set up Black Duck Code Sight.....262
Set up Review Board.....276

Set up the TeamForge Git integration....286
A preface to the TeamForge-Orchestrate integration....287

Frequently asked questions about administrating a TeamForge site.... 288

TeamForge installation FAQs.... 288
Do I need an advanced TeamForge installation?....288
How many servers do I need to run a CollabNet TeamForge site?....289
How does TeamForge handle third-party applications?.... 289
Should I upgrade to TeamForge 7.0 on a new box?.... 290
Should I move my TeamForge database to its own server?....291
Should I move my source control application to its own server?....291
What are the right PostgreSQL settings for my site?.... 291
Is it possible to change artifact prefixes in TeamForge?....292
Can I run other java applications in the same JBoss instance as CollabNet Team Forge?.... 292
What does it mean to run CollabNet TeamForge on a virtual machine?....292
Who is responsible for applying OS updates to the underlying VMware image?.... 292
What is a patch?.... 292
Why do I get a URL "not found" or "moved permanently" error after applying a patch/upgrade?.... 293
Why does the the Yum installer display a warning message on Centos 6?....293
Why am I getting a Yum repository filename conflict?.... 293
When do I run the initial load job?.... 294
Common errors in TeamForge.... 295
Troubleshooting VMware errors.... 295
Troubleshooting database/datamart/ETL errors.... 296
Troubleshooting JBoss errors.... 297
Troubleshooting E-mail errors....298
Troubleshooting other errors.... 299
How does TeamForge manage security?.... 302
What are the minimum ports to keep open for a TeamForge site?....302
How does CollabNet TeamForge help protect data access?....304
What user activities are tracked?.... 304
How does CollabNet TeamForge help protect my data?....305
J2EE Architecture and security....305
What security tools come with CollabNet TeamForge ?....305
What is a CERT advisory?....307
How does TeamForge authenticate CVS users?.... 307
How do I configure Subversion to authenticate against multiple LDAP domains?....308
How do I authenticate multiple LDAP via Apache?.... 309
After switching to ADS authentification, why did the Create button disappear from the user admin section?....309
Does TeamForge work with LDAP?.... 309
Why do I get the "Invalid command 'AuthLDAPAuthoritative'" error when I try to set LDAP for SVN users?....311
How does TeamForge handle multiple redundant LDAP servers?....311
Can the users be forced to change their passwords at first login?....311
Managing email in TeamForge.... 311
How do I configure TeamForge to send mail on a specific network adapter in a multi-NIC configuration?....311
How can I check if port 25 is open?.... 312
How do I set up a local alias via James?.... 312
How do I configure email notifications of Subversion commits in SourceForge 4.x?....313
Does TeamForge support using /etc/aliases for local mail delivery?.... 313
Concepts and terms in TeamForge....313
Advantages of using the Apache TIKA parser library for indexing....313
What is the look project?.... 314
What wiki engine does TeamForge use?.... 314
Does CollabNet TeamForge support merge tracking?....314
What is a private IP address and what are the private IP ranges?....314

What is the vamessages.log used for? 315
How do I use the TeamForge updater to manage backups of old versions of TeamForge? 315
How does TeamForge deliver activity reports? 315
What is an integrated application? 315
How does an integrated application interact with other TeamForge tools? 316
TeamForge roles and permissions 317
Can I set permissions so that users can move documents but not delete them? 317
Why can't Oracle connect to my TeamForge installation? 317
Are role-based permissions allowed for sub-folders in the TeamForge Documents? 318
Can I control user access to an integrated application? 318
Tasks in TeamForge 318
How do I change the time to run the ETL jobs? 318
How can I check the status of ETL? 318
What happens when log files get too big? 318
What is the suggested log configuration for a production system? 319
How do I enable post-commit logging? 319
How do I make the monitoring messages be sent from Forge Administrator? 319
How can I remove the RHEL test page after TeamForge installation? 319
How to reinstall a deleted installation directory? 319
How can I find the number of files in a repository without checking it out? 319
How do I connect to the Datamart? 319
How do I connect to the Teamforge Postgres database? 320
How do I generate a wiki table of contents? 320
What is the correct procedure for modifying a hosted Lab Manager profile? 320
How do I configure the timeout for Apache in TeamForge? 320
How do I back up TeamForge? 320
How do I move an existing CVS repository into TeamForge? 321
How do I move an existing SVN repository into TeamForge? 322
Where do I configure my client proxy settings? 322
How do I make TeamForge work the same when the IP address of the server changes? 323
How do I capture the output of "top" command? 323

Reference information about TeamForge 324
Platform specification for TeamForge 7.0 324
Hardware requirements for CollabNet TeamForge 7.0 324
Software requirements for CollabNet TeamForge 7.0 324
Versions of RPM packages in Red Hat and CentOS installations 326
Versions of RPM packages in SUSE installations 326
Scripts installed with TeamForge 7.0 327
backup-rb-data.py 327
bootstrap-data.sh 328
bootstrap-reporting-data.sh 329
CodeSightMigration.sh 330
The collabnet script 331
datamart-oracle-setup.sh 332
datamart-pgsql-setup.sh 333
db.py 333
domain_change_db.py 333
domain_change_fs.pl 334
domain_change_pt.py 335
environment_check.sh 335
etl-Client.py 335
fix_data_permission.sh 336
install.sh 337
pbl.py 338

password_util.sh....339
psql-wrapper....339
psql-reporting-wrapper....339
restore-data.py....340
SearchReindex.py....340
set_auth_key.py....342
set-reports-readonly-user-permission.py....342
snapshot.py....342
upgrade-site.sh....343
projecttracker.py....344
wmt-wrapper.sh....345
Log files in TeamForge 7.0 on Red HatCentOSSuSEVMware Player....346
JBoss logs....346
Oracle logging....346
SCM (CVS, Subversion, and Perforce) logs....347
Email logs....347
Search logs....348
Project Build Library audit log....348
Profile audit log....349
User Audit Log....349
Host audit log....349
Project audit log....350
etl.log....350
Configuration files in TeamForge 7.0 on Red HatCentOSSuSEVMware Player....350
site-options.conf....350
c6migrate.conf variables....388
httpd.conf....388
pebble-app.xml....389
pebble-dep.xml....391
How is an integrated application described?....392
install.conf....395
install.conf....395
iptables....396
The patch manifest file....396
postresql.conf....397

CollabNet TeamForge 7.0 release notes....399
New features in CollabNet TeamForge 7.0....399
Issues resolved in CollabNet TeamForge 7.0....402
Known issues in CollabNet TeamForge 7.0....402

CollabNet TeamForge 7.0 Patch 1 release notes....406

TeamForge system administrator how-tos

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

Install CollabNet TeamForge 7.0

When you finish this sequence of tasks, you will have a working TeamForge 7.0 site tailored to the specific requirements of your user base and environment.

Note: TeamForge 7.0 Patch 1 is available. *Click here to see the release notes*. By default, TeamForge 7.0 Patch 1 is installed when you install TeamForge 7.0. However, you can downgrade to TeamForge 7.0 if you choose to, by following the help topics given below.

- *Downgrade a patch on RHEL / CentOS* on page 227
- *Downgrade a patch on SuSE* on page 229

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 *Platform specification for TeamForge 7.0* on page 324.

Plan your CollabNet TeamForge 7.0 installation

Before you install TeamForge 7.0, 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.
- The Extract transform and load (ETL) server pulls data from the site database and populates the datamart to generate charts and graphs about how people are using the site.
- The datamart is an abstraction of the site database, optimized to support the reporting functionality.

Important: To activate Black Duck Code Sight, a dedicated Code Sight licensing key is required. To receive a Code Sight license key, contact your CollabNet account manager, or send an email request to info@collab.net. (A commercial TeamForge license is required, i.e. TeamForge "free option license" does not qualify).

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.

PostgreSQL or Oracle?

PostgreSQL 9.0.13 is installed automatically when you install TeamForge 7.0. Oracle 11G (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 7.0* on page 324.

Set up networking for your TeamForge box

After installing the operating system, prepare the networking connections and configuration that your TeamForge 7.0 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 302
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>
   export no_proxy=localhost,127.0.0.0/8,<hostname>
   ```
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 302 for detailed security recommendations.)

Install CollabNet TeamForge 7.0 on Red Hat/CentOS

Install TeamForge the dedicated way

Dedicated installation mode is suitable when the server is fully dedicated to TeamForge application. This makes the installation process very simple as the TeamForge installer takes care of configuring your apache, postgresql, selinux (if enabled) automatically without any manual intervention

In a "Dedicated" install, it is advised not to install another software application other than Teamforge. It is also recommended not to run any unwanted services that are not used by TeamForge.

Install TeamForge 7.0 with all services on the same box

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: There are no changes in the steps for installing TeamForge 7.0 Patch 1

In TeamForge 7.0, this scenario includes installing Black Duck Code Sight and Git Integration as well on this box.

Important: It is *critical* that you start with a fresh machine, without any software installed. You must have root access to the server.

1. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.
   - The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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 7 for details.
3. Set the TeamForge repository configuration.

   If you are installing TeamForge with internet access:
   - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
   - Copy it to `/etc/yum.repos.d/`.
   - Refresh your repository cache.

     ```
     yum clean all
     ```

   If you are installing TeamForge without internet access:

   a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

   b) Unpack the disconnected installation package.

   ```
   rpm -ivh <package-name>
   ```

   c)

   Note: If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

   Insert the Red Hat installation DVD.

   This will provide some of the necessary software and utilities for installing TeamForge.

   d) Mount the Red Hat installation DVD.

```
cd /media/
mkdir cdrom
mount /dev/cdrom ./cdrom/
```

Replace `cdrom` with the identifier for your server's CD drive, if necessary.

Tip: If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

4. Install the following application packages.

a) TeamForge: To install the TeamForge application packages run the following command:

```
yum install teamforge
```

b) GIT: To install the GIT packages run the following command:

```
yum install teamforge-git
```

c) Black Duck Code Sight: To install Black Duck Code Sight run the following command:

```
yum install teamforge-codesearch
```

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

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

At this stage, we'll make the minimal modifications required to set up a usable TeamForge 7.0 site.

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

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) Configure the TeamForge services to run in the local box.

```
HOST_localhost=app etl database datamart cvs subversion indexer
```

c) Review the *DATABASE* variables.

You may want to substitute your own values for the default database names, user names and passwords.

d) Turn on the site-wide reporting functionality by adding the *REPORTS_** variables.

See *Turn on site-wide reporting* on page 250 for details.

e) By default, the site wide reports are not commented. To turn on the site wide reports, set the values. Else, comment out the following reports.

- REPORTS_DATABASE_NAME=
- REPORTS_DATABASE_PASSWORD=
- REPORTS_DATABASE_USERNAME=
- REPORTS_DATABASE_READ_ONLY_USER=
- REPORTS_DATABASE_READ_ONLY_PASSWORD=

f) **It is mandatory to include the *SCM_DEFAULT_SHARED_SECRET* token in the `site-options.conf` file of the primary TeamForge server, and provide it with a value of 16-24 characters. Remember to use the same key in the external SCM integration server also.**

g) If the token *REQUIRE_RANDOM_ADMIN_PASSWORD* is already set to true, then set the token *ADMIN_EMAIL* with a valid email address.

```
ADMIN_EMAIL=root@{__APPLICATION_HOST__}
```

h) **Important:** If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

i) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

j) Include the list of TeamForge user accounts whose password should not expire.

```
USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer
```

If you are installing git integration, add a TeamForge user for example,"gitadmin" with site-administrator rights and append the username against this parameter.

```
USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer,gitadmin
```

k) Configure the following settings if you are installing Git:

```
HOST_localhost=app etl database datamart subversion cvs indexer gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

l) Configure the following settings for Black Duck Code Sight.

```
HOST_localhost=app etl database datamart subversion cvs
 indexer codesearch
```

Note: In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>

To enable SSL for Black Duck Code Sight, include this token:
BDCS_SSL=on
```

Note: If you have Black Duck Code Sight on SSL and the following external certificate tokens are not provided, the installer will generate a self-signed certificate.

```
For valid SSL certificates, configure the following tokens:
BDCS_SSL_CERT_FILE=
BDCS_SSL_KEY_FILE=
```

```
The ca.crt and chain files are optional -- leave out the tokens if you
 don't use the files.
BDCS_SSL_CA_CERT_FILE=
BDCS_SSL_CHAIN_FILE=
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
BDCS_ADMIN_PASSWORD=<password>
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

Advanced Black Duck Code Sight configuration settings:

Note: The following are the advanced configuration tokens which can be set once during the time of installation.

```
The path where the repositories enabled for codesearch to check out.
BDCS_SCAN_SOURCE_DIR_ROOT=/opt/collabnet/blackduck/scan

The path where the codesearch software is installed.
BDCS_INSTALL_PATH=/opt/collabnet/blackduck

The path where codesearch database is installed.
BDCS_PGSQL_HOME_DIR_ROOT=/opt/collabnet/blackduck/postgres

The port number for codesearch db server.
BDCS_PGSQL_PORT=55435

The tomcat maximum heap memory size in megabytes.
BDCS_TOMCAT_MX_IN_MB=1024

The shutdown port number for codesearch tomcat server.
BDCS_TOMCAT_SHUTDOWN_PORT=9189
```

6. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

7. Set up the initial site data (bootstrap).

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

8. Start TeamForge.

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

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

9. If you have installed Git integration, add the "gitadmin" user with site-administrator rights through TeamForge user interface.

10. If you have installed Black Duck Code Sight, then install the license for Black Duck Code Sight. For more information, see *Install the Black Duck Code Sight license* on page 274.
11. Integrate gerrit by running the `post-install.py` script.

    Tip: Skip this step if you are not installing git integration.

    ```
    /opt/collabnet/gerrit/scripts/post-install.py
    ```

    The post installation script will try to detect the required configuration parameters. The following 3 parameters do not have default values and you will be asked to provide them:

    - Teamforge Login name: the dedicated TeamForge site administrator account that does not expire and cannot be locked
    - Teamforge Password: the password for the above account
    - Database password: the password to protect Gerrit's database from unauthorized access. Specify its value when you first run the post-install script. Make sure you note the value because you will be asked for it later.

    a) Restart the Gerrit services.

    ```
    /etc/init.d/collabnet restart gerrit
    ```

    b) To verify the Git integration, run the following command.

    ```
    /etc/init.d/collabnet status
    ```

12. Run the following initial ETL load jobs.

    a) Change to the `runtime/scripts` directory.

    ```
    cd /opt/collabnet/teamforge/runtime/scripts
    ```

    b) Run the `TrackerInitialJob`.

    ```
    ./etl-client.py -r TrackerInitialJob
    ```

    c) Run the `SCMInitialJob`.

    ```
    ./etl-client.py -r SCMCommitInitialJob
    ```

    Tip: For more information see *When do I run the initial load job?* on page 294.

13. Run the `post-install.py` script after the TeamForge application is up and running.

    ```
    sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
    ```

14. To integrate the Black Duck Code Sight with the TeamForge run the Black Duck Code Sight `post-install.sh` script.

    Note: It is assumed that Subversion's client configuration file (`/root/.subversion/config`) for the root user is the default one without customization.

    ```
    sudo /opt/collabnet/teamforge/runtime/scripts/codesearch/post-install.sh
    ```

15. If the token *VALIDATE_SSL_CERTS* is set to "true", you must run the codesearch runtime script *trust-cert.sh* in the application box.

    ```
    cd /opt/collabnet/teamforge/runtime/scripts/codesearch
    ./trust-cert.sh
    ```

16. Restart the `collabnet` services.

    ```
    /etc/init.d/collabnet restart
    ```

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

    a) 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.

    b) Create a sample project.

    See *Create a TeamForge project*.

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

See *Create a site-wide broadcast*.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Install TeamForge 7.0 with database and datamart on a separate box

In this option, we install database and datamart (reporting database) on a separate box and other services on the main application box.

Important: There are no changes in the steps for installing TeamForge 7.0 Patch 1

We call this a "dedicated install" since both the servers are dedicated to TeamForge. TheTeamForge installer takes care of the database configurations. 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.

Important: If the remote machine where the datamart is running is not under your direct control, check with the database administrator to make sure that you can carry out these instructions on that box.

Note: In a multi-box installation of TeamForge, ensure that all servers have the same system time zone for ETL to function properly.

Do this on my.app.box

1. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.
   - The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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 7 for details.
3. Set the TeamForge repository configuration.

   If you are installing TeamForge with internet access:

   - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
   - Copy it to `/etc/yum.repos.d/`.
   - Refresh your repository cache.

     ```
     yum clean all
     ```

   If you are installing TeamForge without internet access:

   a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

   b) Unpack the disconnected installation package.

   ```
   rpm -ivh <package-name>
   ```

   c) **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

   Insert the Red Hat installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) Mount the Red Hat installation DVD.

```
cd /media/
mkdir cdrom
mount /dev/cdrom ./cdrom/
```

Replace `cdrom` with the identifier for your server's CD drive, if necessary.

Tip: If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

4. Install the following TeamForge application, ETL and source control services.

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

Tip: If the yum installer fails, you may have duplicate rpm packages installed.

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 teamforge-scm
   ```

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

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

At this stage, we'll make the minimal modifications required to set up a usable TeamForge 7.0 site.

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

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) Identify the boxes and the services running on them.

```
HOST_localhost=app etl subversion cvs indexer
DOMAIN_localhost=worlddomination.supervillain.org
HOST_<my.databaseanddatamart.box>=database datamart
```

c) Configure the database and datamart.

Tip: For more information about configuring variables, see *site-options.conf* on page 350.

```
DATABASE_TYPE=postgresql
DATABASE_USERNAME=ctfuser
DATABASE_PASSWORD=ctfpwd
DATABASE_NAME=ctfdb
DATABASE_MAX_POOL_SIZE=220
DATABASE_READ_ONLY_USER=ctfrouser
DATABASE_READ_ONLY_PASSWORD=ctfropwd
REPORTS_DATABASE_USERNAME=ctfrptuser
REPORTS_DATABASE_PASSWORD=ctfrptpwd
REPORTS_DATABASE_NAME=ctfrptdb
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

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

We also recommend the following:

- For small dataset customers: use the same instance as that of `ctfdb`; do not specify REPORTS_DATABASE_PORT.
- For medium to large dataset customers: use a separate instance; specify REPORTS_DATABASE_PORT.

d)

Important: If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

e) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

f) **It is mandatory to include the *SCM_DEFAULT_SHARED_SECRET* token in the `site-options.conf` file of the primary TeamForge server, and provide it with a value of 16-24 characters. Remember to use the same key in the external SCM integration server also.**

6. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

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

```
export http_proxy=
```

Do this on the database or datamart server. We'll call this my.databaseanddatamart.box

8. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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.

9. Install the TeamForge database packages.

```
yum install teamforge-database
```

10. Copy the `site-options.conf` file from the application box to the `/opt/collabnet/teamforge-installer/7.0.0.1/conf` directory on the my.databaseanddatamart.box.
11. Modify the host token settings on the `site-options.conf` file.

```
HOST_localhost=database datamart
DOMAIN_localhost=db.supervillain.org
HOST_<my.app.box>=app etl subversion cvs indexer
```

12. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

Do this on my.app.box:

13. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./bootstrap-data.sh
```

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.

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

15. Integrate the CLI reports by running the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

16. Run the following initial ETL load jobs.
    a) Change to the `runtime/scripts` directory.

    ```
    cd /opt/collabnet/teamforge/runtime/scripts
    ```

    b) Run the `TrackerInitialJob`.

    ```
    ./etl-client.py -r TrackerInitialJob
    ```

    c) Run the `SCMInitialJob`.

    ```
    ./etl-client.py -r SCMCommitInitialJob
    ```

Tip: For more information see *When do I run the initial load job?* on page 294.

17. Revoke the super user permissions of the database and datamart users.

```
/opt/collabnet/teamforge/runtime/scripts/revoke-superuser-permission.py
```

Do this in my.databaseanddatamart.box

18. Run the following script to set permissions for the TeamForge database read-only user specified by the *DATABASE_READ_ONLY_USER* token.

```
/opt/collabnet/teamforge/runtime/scripts/set-ctfdb-readonly-user-permission.py
```

19. Run the following script to set permissions for the reporting database read-only user.

```
/opt/collabnet/teamforge/runtime/scripts/set-reports-readonly-user-permission.py
```

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Install TeamForge 7.0 with reporting services on a separate box

In this option, we install the datamart and ETL on a separate box and other services on the main application box.

Important: There are no changes in the steps for installing TeamForge 7.0 Patch 1

Note: In a multi-box installation of TeamForge, all machines must have the same time - without this, ETL will not function properly.

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

1. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.
   - The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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 7 for details.

3. Set the TeamForge repository configuration.

   If you are installing TeamForge with internet access:

   - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
   - Copy it to `/etc/yum.repos.d/`.
   - Refresh your repository cache.

     ```
     yum clean all
     ```

   If you are installing TeamForge without internet access:

   a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

   b) Unpack the disconnected installation package.

   ```
   rpm -ivh <package-name>
   ```

   c) **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

   Insert the Red Hat installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) Mount the Red Hat installation DVD.

```
cd /media/
mkdir cdrom
mount /dev/cdrom ./cdrom/
```

Replace `cdrom` with the identifier for your server's CD drive, if necessary.

Tip: If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

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

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

Tip: If the yum installer fails, you may have duplicate rpm packages installed.

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 teamforge-scm
   ```

5. Install the following application packages.

a) GIT: To install the GIT packages run the following command.

```
yum install teamforge-git
```

b) To install Black Duck Code Sight run the following command.

```
yum install teamforge-codesearch
```

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

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

At this stage, we'll make the minimal modifications required to set up a usable TeamForge 7.0 site.

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

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) Identify the boxes and the services running on them.

```
HOST_localhost=app database subversion cvs indexer
DOMAIN_localhost=worlddomination.supervillain.org
HOST_<my.datamartetl.box>=datamart etl
```

c) Review the *DATABASE* variables.

You may want to substitute your own values for the default database names, user names and passwords.

d) Turn on the site-wide reporting functionality by adding the *REPORTS_** variables.

See *Turn on site-wide reporting* on page 250 for details.

e) By default, the site wide reports are not commented. To turn on the site wide reports, set the values. Else, comment out the following reports.

- REPORTS_DATABASE_NAME=
- REPORTS_DATABASE_PASSWORD=
- REPORTS_DATABASE_USERNAME=
- REPORTS_DATABASE_READ_ONLY_USER=
- REPORTS_DATABASE_READ_ONLY_PASSWORD=

f) **It is mandatory to include the *SCM_DEFAULT_SHARED_SECRET* token in the `site-options.conf` file of the primary TeamForge server, and provide it with a value of 16-24 characters. Remember to use the same key in the external SCM integration server also.**

g) If the token *REQUIRE_RANDOM_ADMIN_PASSWORD* is already set to true, then set the token *ADMIN_EMAIL* with a valid email address.

```
ADMIN_EMAIL=root@{__APPLICATION_HOST__}
```

h) **Important:** If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

i) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

j) If you're installing Git, configure the following settings:

```
HOST_localhost=app database subversion cvs indexer gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

k) Configure the following settings for Black Duck Code Sight.

```
HOST_localhost=app etl database datamart subversion cvs
 indexer codesearch
```

Note: In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>
```

```
To enable SSL for Black Duck Code Sight, include this token:
BDCS_SSL=on
```

👉 **Note:** If you have Black Duck Code Sight on SSL and the following external certificate tokens are not provided, the installer will generate a self-signed certificate.

```
For valid SSL certificates, configure the following tokens:
BDCS_SSL_CERT_FILE=
BDCS_SSL_KEY_FILE=
The ca.crt and chain files are optional -- leave out the tokens if you
 don't use the files.
BDCS_SSL_CA_CERT_FILE=
BDCS_SSL_CHAIN_FILE=
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
BDCS_ADMIN_PASSWORD=<password>
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

Advanced Black Duck Code Sight configuration settings:

👉 **Note:** The following are the advanced configuration tokens which can be set once during the time of installation.

```


The path where the repositories enabled for codesearch to check out.
BDCS_SCAN_SOURCE_DIR_ROOT=/opt/collabnet/blackduck/scan

The path where the codesearch software is installed.
BDCS_INSTALL_PATH=/opt/collabnet/blackduck

The path where codesearch database is installed.
BDCS_PGSQL_HOME_DIR_ROOT=/opt/collabnet/blackduck/postgres

The port number for codesearch db server.
BDCS_PGSQL_PORT=55435

The tomcat maximum heap memory size in megabytes.
BDCS_TOMCAT_MX_IN_MB=1024

The shutdown port number for codesearch tomcat server.
BDCS_TOMCAT_SHUTDOWN_PORT=9189
```

7. Create the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

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

```
export http_proxy=
```

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

9. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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 the TeamForge database and ETL packages.

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

11. Copy the `site-options.conf` file from the application box to the `/opt/collabnet/teamforge-installer/7.0.0.1/conf` directory on the my.datamartetl.box.
12. Modify the host token settings on the `site-options.conf` file.

```
HOST_localhost=datamart etl
DOMAIN_localhost=etl.supervillain.org
HOST_<my.app.box>=app database subversion cvs indexer
```

13. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

Do this on my.app.box.

14. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./bootstrap-data.sh
```

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.

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

16. If you have installed Black Duck Code Sight, then install the license for Black Duck Code Sight. For more information, see *Install the Black Duck Code Sight license* on page 274.
17. Integrate gerrit by running the `post-install.py` script.

Tip: Skip this step if you are not installing git integration.

```
/opt/collabnet/gerrit/scripts/post-install.py
```

The post installation script will try to detect the required configuration parameters. The following 3 parameters do not have default values and you will be asked to provide them:

- Teamforge Login name: the dedicated TeamForge site administrator account that does not expire and cannot be locked

- Teamforge Password: the password for the above account
- Database password: the password to protect Gerrit's database from unauthorized access. Specify its value when you first run the post-install script. Make sure you note the value because you will be asked for it later.

a) Restart the Gerrit services.

```
/etc/init.d/collabnet restart gerrit
```

b) To verify the Git integration, run the following command.

```
/etc/init.d/collabnet status
```

18. Integrate the CLI reports by running the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

19. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

20. Integrate Black Duck Code Sight with TeamForge by running the Black Duck Code Sight `post-install.sh` script.

Note: It is assumed that Subversion's client configuration file (`/root/.subversion/config`) for the root user is the default one without customization.

Note: If the Black Duck Code Sight is running on a separate server, run the following command in the code sight server.

```
sudo /opt/collabnet/teamforge/runtime/scripts/codesearch/post-install.sh
```

21. After you run the Black Duck Code Sight `post-install.sh` script, run the following script from the same directory:

```
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL similar to "http://myint.box.net/svn/repos" where `myint.box` is the box with the SCM integration server.

Also, if you add a new integration server at some point later, you'll need to run this script, (after creating new integration server), in the TeamForge `app` box.

Important: Do this if you have a multi-box setup with a SCM integration server on a separate server.

22. If the token *VALIDATE_SSL_CERTS* is set to "true", you must run the codesearch runtime script *trust-cert.sh* in the application box.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
```

23. Restart the `collabnet` services.

```
/etc/init.d/collabnet restart
```

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

a) 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.

b) Create a sample project.

See *Create a TeamForge project*.

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

See *Create a site-wide broadcast*.

25. Revoke the super user permissions of database and datamart users.

```
/opt/collabnet/teamforge/runtime/scripts/revoke-superuser-permission.py
```

26. Run the following script to set permissions for the reporting database read-only user.

```
/opt/collabnet/teamforge/runtime/scripts/set-reports-readonly-user-
permission.py
```

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Install TeamForge 7.0 with Black Duck Code Sight on a separate box on Red Hat/CentOS

In this option, we install Black Duck Code Sight on a separate box on Red Hat Enterprise Linux and other services on the main application box.

Important: There are no changes in the steps for installing TeamForge 7.0 Patch 1

Note: In a multi-box installation of TeamForge, all machines must have the same time zone - without this, ETL will not function properly.

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

1. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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 7 for details.

3. Set the TeamForge repository configuration.

If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
yum clean all
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

Insert the Red Hat installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) Mount the Red Hat installation DVD.

```
cd /media/
mkdir cdrom
mount /dev/cdrom ./cdrom/
```

Replace `cdrom` with the identifier for your server's CD drive, if necessary.

Tip: If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

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

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

Tip: If the yum installer fails, you may have duplicate rpm packages installed.

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 teamforge-scm
   ```

5. Install the following application packages.

a) GIT: To install the GIT packages run the following command.

```
yum install teamforge-git
```

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

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

At this stage, we'll make the minimal modifications required to set up a usable TeamForge 7.0 site.

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

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) Configure the TeamForge services to run in the local box.

```
HOST_localhost=app etl database datamart cvs subversion indexer
```

c) Review the *DATABASE* variables.

You may want to substitute your own values for the default database names, user names and passwords.

d) Turn on the site-wide reporting functionality by adding the *REPORTS_** variables.

See *Turn on site-wide reporting* on page 250 for details.

e) By default, the site wide reports are not commented. To turn on the site wide reports, set the values. Else, comment out the following reports.

- REPORTS_DATABASE_NAME=
- REPORTS_DATABASE_PASSWORD=
- REPORTS_DATABASE_USERNAME=
- REPORTS_DATABASE_READ_ONLY_USER=
- REPORTS_DATABASE_READ_ONLY_PASSWORD=

f) **It is mandatory to include the *SCM_DEFAULT_SHARED_SECRET* token in the `site-options.conf` file of the primary TeamForge server, and provide it with a value of 16-24 characters. Remember to use the same key in the external SCM integration server also.**

g) If the token *REQUIRE_RANDOM_ADMIN_PASSWORD* is already set to true, then set the token *ADMIN_EMAIL* with a valid email address.

```
ADMIN_EMAIL=root@{__APPLICATION_HOST__}
```

h) **Important:** If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

i) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

j) Configure the following settings if you are installing Git:

```
HOST_localhost=app etl database datamart subversion cvs indexer gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

k) Configure the following settings for Black Duck Code Sight.

```
HOST_my.codesight.hostname=codesearch
```

Note: In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>

To enable SSL for Black Duck Code Sight, include this token:
BDCS_SSL=on
```

Note: If you have Black Duck Code Sight on SSL and the following external certificate tokens are not provided, the installer will generate a self-signed certificate.

```
For valid SSL certificates, configure the following tokens:
BDCS_SSL_CERT_FILE=
BDCS_SSL_KEY_FILE=
The ca.crt and chain files are optional -- leave out the tokens if you
 don't use the files.
BDCS_SSL_CA_CERT_FILE=
BDCS_SSL_CHAIN_FILE=
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
BDCS_ADMIN_PASSWORD=<password>
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

Advanced Black Duck Code Sight configuration settings:

Note: The following are the advanced configuration tokens which can be set once during the time of installation.

```


The path where the repositories enabled for codesearch to check out.
BDCS_SCAN_SOURCE_DIR_ROOT=/opt/collabnet/blackduck/scan

The path where the codesearch software is installed.
BDCS_INSTALL_PATH=/opt/collabnet/blackduck

The path where codesearch database is installed.
BDCS_PGSQL_HOME_DIR_ROOT=/opt/collabnet/blackduck/postgres

The port number for codesearch db server.
BDCS_PGSQL_PORT=55435

The tomcat maximum heap memory size in megabytes.
BDCS_TOMCAT_MX_IN_MB=1024

The shutdown port number for codesearch tomcat server.
BDCS_TOMCAT_SHUTDOWN_PORT=9189
```

7. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

Note: If you are setting up CollabNet TeamForge on a machine that is meant for emergency backup or disaster recovery, use the -R option in place of the -r option.

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

```
export http_proxy=
```

Do this on the Code Sight server. We'll call this my.codesight.box

9. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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. Run the following command to install Black Duck Code Sight.

```
yum install teamforge-codesearch
```

11. Copy the `site-options.conf` file from the application box to the code search box in the directory `/opt/collabnet/teamforge-installer/7.0.0.1/conf`
12. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

Do this on my.app.box:

13. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./bootstrap-data.sh
```

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.

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

15. Install the Black Duck Code Sight license on the box where Black Duck Code Sight is installed. For more information, see *Install the Black Duck Code Sight license* on page 274.
16. Integrate gerrit by running the `post-install.py` script.

Tip: Skip this step if you are not installing git integration.

```
/opt/collabnet/gerrit/scripts/post-install.py
```

The post installation script will try to detect the required configuration parameters. The following 3 parameters do not have default values and you will be asked to provide them:

- Teamforge Login name: the dedicated TeamForge site administrator account that does not expire and cannot be locked
- Teamforge Password: the password for the above account
- Database password: the password to protect Gerrit's database from unauthorized access. Specify its value when you first run the post-install script. Make sure you note the value because you will be asked for it later.

a) Restart the Gerrit services.

```
/etc/init.d/collabnet restart gerrit
```

b) To verify the Git integration, run the following command.

```
/etc/init.d/collabnet status
```

17. Integrate the CLI reports by running the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

18. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

Do this in my.codesight.box

19. To integrate Black Duck Code Sight with TeamForge run the Black Duck Code Sight `post-install.sh` script.

Note: It is assumed that Subversion's client configuration file (`/root/.subversion/config`) for the root user is the default one without customization.

```
sudo /opt/collabnet/teamforge/runtime/scripts/codesearch/post-install.sh
```

20. If the token *VALIDATE_SSL_CERTS* is set to "true", you must run the codesearch runtime script *trust-cert.sh* in the application box.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
```

Do this on my.app.box

21. Restart the TeamForge services.

```
/etc/init.d/collabnet restart tomcatcs
```

Do this on my.codesight.box

22. Restart the Black Duck Code Sight services.

```
/etc/init.d/collabnet restart
```

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

a) 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.

b) Create a sample project.

See *Create a TeamForge project*.

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

See *Create a site-wide broadcast*.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Install TeamForge 7.0 with Git integration on a separate box

In this option, we install Git integration on a separate box and other services on the main application box.

Important: There are no changes in the steps for installing TeamForge 7.0 Patch 1

Note: In a multi-box installation of TeamForge, all machines must have the same time zone - without this, ETL will not function properly.

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

1. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.
   - The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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 7 for details.

3. Set the TeamForge repository configuration.

   If you are installing TeamForge with internet access:

   - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
   - Copy it to `/etc/yum.repos.d/`.
   - Refresh your repository cache.

     ```
     yum clean all
     ```

   If you are installing TeamForge without internet access:

   a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

   b) Unpack the disconnected installation package.

   ```
   rpm -ivh <package-name>
   ```

   c) **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

   Insert the Red Hat installation DVD.

   This will provide some of the necessary software and utilities for installing TeamForge.

   d) Mount the Red Hat installation DVD.

   ```
   cd /media/
   mkdir cdrom
   mount /dev/cdrom ./cdrom/
   ```

   Replace `cdrom` with the identifier for your server's CD drive, if necessary.

   Tip: If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

   e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

   f) Copy this into the yum configuration file:

   ```
   [RHEL-CDROM]
   name=RHEL CDRom
   baseurl=file:///media/cdrom/Server/
   gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
   enabled=1
   gpgcheck=0
   ```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

4. Install the following application packages.
   a) TeamForge: To install the TeamForge application packages run the following command:

   ```
   yum install teamforge
   ```

   b) Black Duck Code Sight: To install Black Duck Code Sight run the following command:

   ```
   yum install teamforge-codesearch
   ```

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

   ```
   vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
   ```

   At this stage, we'll make the minimal modifications required to set up a usable TeamForge 7.0 site.

   Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

   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) Identify the boxes and the services running on them.

   ```
   HOST_localhost=app database subversion cvs datamart etl indexer
   DOMAIN_localhost=worlddomination.supervillain.org
   HOST_<my.git.box>=gerrit
   ```

   c) Review the *DATABASE* variables.
   You may want to substitute your own values for the default database names, user names and passwords.
   d) Turn on the site-wide reporting functionality by adding the *REPORTS_** variables.
   See *Turn on site-wide reporting* on page 250 for details.
   e) By default, the site wide reports are not commented. To turn on the site wide reports, set the values. Else, comment out the following reports.

   - REPORTS_DATABASE_NAME=
   - REPORTS_DATABASE_PASSWORD=
   - REPORTS_DATABASE_USERNAME=
   - REPORTS_DATABASE_READ_ONLY_USER=
   - REPORTS_DATABASE_READ_ONLY_PASSWORD=

   f) **It is mandatory to include the *SCM_DEFAULT_SHARED_SECRET* token in the `site-options.conf` file of the primary TeamForge server, and provide it with a value of 16-24 characters. Remember to use the same key in the external SCM integration server also.**
   g) Include the list of TeamForge user accounts for the tokens whose password should not expire in the comma separated format.

   ```
   USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer
   ```

If you are installing git integration, add a TeamForge user viz.,"gitadmin" with site-administrator rights and append the username against this parameter.

```
USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer,gitadmin
```

h) If the token *REQUIRE_RANDOM_ADMIN_PASSWORD* is already set to true, then set the token *ADMIN_EMAIL* with a valid email address.

```
ADMIN_EMAIL=root@{__APPLICATION_HOST__}
```

i) 👉 **Important:** If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

j) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

k) Configure the following settings for Black Duck Code Sight.

```
HOST_localhost=app etl database datamart subversion cvs
 indexer codesearch
```

👉 **Note:** In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>

To enable SSL for Black Duck Code Sight, include this token:
BDCS_SSL=on
```

👉 **Note:** If you have Black Duck Code Sight on SSL and the following external certificate tokens are not provided, the installer will generate a self-signed certificate.

```
For valid SSL certificates, configure the following tokens:
BDCS_SSL_CERT_FILE=
BDCS_SSL_KEY_FILE=
The ca.crt and chain files are optional -- leave out the tokens if you
 don't use the files.
BDCS_SSL_CA_CERT_FILE=
BDCS_SSL_CHAIN_FILE=
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
BDCS_ADMIN_PASSWORD=<password>
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

Advanced Black Duck Code Sight configuration settings:

👉 **Note:** The following are the advanced configuration tokens which can be set once during the time of installation.

```
The path where the repositories enabled for codesearch to check out.
BDCS_SCAN_SOURCE_DIR_ROOT=/opt/collabnet/blackduck/scan
```

```
The path where the codesearch software is installed.
BDCS_INSTALL_PATH=/opt/collabnet/blackduck

The path where codesearch database is installed.
BDCS_PGSQL_HOME_DIR_ROOT=/opt/collabnet/blackduck/postgres

The port number for codesearch db server.
BDCS_PGSQL_PORT=55435

The tomcat maximum heap memory size in megabytes.
BDCS_TOMCAT_MX_IN_MB=1024

The shutdown port number for codesearch tomcat server.
BDCS_TOMCAT_SHUTDOWN_PORT=9189
```

6. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

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

```
export http_proxy=
```

8. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./bootstrap-data.sh
```

9. 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.

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

10. If you are installing git integration, add the "gitadmin" user with site-administrator rights through TeamForge user interface.

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

11. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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.

12. Run the following command to install the Git packages.

```
yum install teamforge-git
```

13. Copy the `site-options.conf` from the application box.
14. If you're installing Git, configure the following settings:

```
HOST_my.git=gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

15. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

16. Integrate gerrit by running the `post-install.py` script.

Tip: Skip this step if you are not installing git integration.

```
/opt/collabnet/gerrit/scripts/post-install.py
```

The post installation script will try to detect the required configuration parameters. The following 3 parameters do not have default values and you will be asked to provide them:

- Teamforge Login name: the dedicated TeamForge site administrator account that does not expire and cannot be locked
- Teamforge Password: the password for the above account
- Database password: the password to protect Gerrit's database from unauthorized access. Specify its value when you first run the post-install script. Make sure you note the value because you will be asked for it later.

a) Restart the Gerrit services.

```
/etc/init.d/collabnet restart gerrit
```

b) To verify the Git integration, run the following command.

```
/etc/init.d/collabnet status
```

Do this in my.app.box

17. If you have installed Black Duck Code Sight, then install the license for Black Duck Code Sight. For more information, see *Install the Black Duck Code Sight license* on page 274.
18. Integrate the CLI reports by running the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

19. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

20. Integrate Black Duck Code Sight with TeamForge by running the Black Duck Code Sight `post-install.sh` script.

Note: It is assumed that Subversion's client configuration file (`/root/.subversion/config`) for the root user is the default one without customization.

Note: If the Black Duck Code Sight is running on a separate server, run the following command in the code sight server.

```
sudo /opt/collabnet/teamforge/runtime/scripts/codesearch/post-install.sh
```

21. After you run the Black Duck Code Sight `post-install.sh` script, run the following script from the same directory:

    ```
    ./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
    ```

    Provide a repository base URL similar to "http://myint.box.net/svn/repos" where `myint.box` is the box with the SCM integration server.

    Also, if you add a new integration server at some point later, you'll need to run this script, (after creating new integration server), in the TeamForge `app` box.

    Important: Do this if you have a multi-box setup with a SCM integration server on a separate server.

22. If the token *VALIDATE_SSL_CERTS* is set to "true", you must run the codesearch runtime script *trust-cert.sh* in the application box.

    ```
    cd /opt/collabnet/teamforge/runtime/scripts/codesearch
    ./trust-cert.sh
    ```

23. Restart the `collabnet` services.

    ```
    /etc/init.d/collabnet restart
    ```

24. Apply some finishing touches and make sure everything is running smoothly.
    a) 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.
    b) Create a sample project.

       See *Create a TeamForge project*.
    c) Write a welcome message to your site's users.

       See *Create a site-wide broadcast*.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Install TeamForge 7.0 on three boxes for intensive database utilization

In this option, we install the database and datamart (reporting database) on one box, scm (cvs/subversion/git) on the second box and other services on the main application box. This may be appropriate when intensive database utilization is a factor.

Important: There are no changes in the steps for installing TeamForge 7.0 Patch 1

Note: If the remote machine in which the datamart is running 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.

Note: In a multi-box installation of TeamForge, all machines must have the same time zone - without this, ETL will not function properly.

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

1. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.
   - The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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 7 for details.
3. Set the TeamForge repository configuration.

If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
yum clean all
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

Insert the Red Hat installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) Mount the Red Hat installation DVD.

```
cd /media/
mkdir cdrom
mount /dev/cdrom ./cdrom/
```

Replace `cdrom` with the identifier for your server's CD drive, if necessary.

Tip: If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

4. Install the TeamForge application, reporting service.

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

Tip: If the yum installer fails, you may have duplicate rpm packages installed.

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. Set up your site's master configuration file.

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

At this stage, we'll make the minimal modifications required to set up a usable TeamForge 7.0 site.

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

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) Identify the servers and the services running on them.

```
HOST_localhost=app indexer etl
DOMAIN_localhost=<myappbox.domain.com>
HOST_<my.databaseanddatamart.box>=database datamart
HOST_<my.scmandgit.box>=subversion cvs gerrit
```

c) Configure the database and the datamart.

Tip: For more information about configuring variables, see *site-options.conf* on page 350.

```
DATABASE_TYPE=postgresql
DATABASE_USERNAME=ctfuser
DATABASE_PASSWORD=ctfpwd
DATABASE_NAME=ctfdb
DATABASE_READ_ONLY_USER=ctfrouser
DATABASE_READ_ONLY_PASSWORD=ctfropwd
DATABASE_MAX_POOL_SIZE=220
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
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

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

We also recommend the following:

- For small dataset customers: use the same instance as that of `ctfdb`; do not specify REPORTS_DATABASE_PORT.
- For medium to large dataset customers: use a separate instance; specify REPORTS_DATABASE_PORT.

d) **Important:** If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

e) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

f) **It is mandatory to include the *SCM_DEFAULT_SHARED_SECRET* token in the `site-options.conf` file of the primary TeamForge server, and provide it with a value of 16-24 characters. Remember to use the same key in the external SCM integration server also.**

6. Create the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

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

```
export http_proxy=
```

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

8. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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.

9. Install the TeamForge database packages.

```
yum install teamforge-database
```

10. Copy the `site-options.conf` file from the application box to the `/opt/collabnet/teamforge-installer/7.0.0.1/conf` directory on the my.databaseanddatamart.box.

11. Modify the host token settings on the `site-options.conf` file.

```
HOST_localhost=database datamart
DOMAIN_localhost=<db.domain.com>
HOST_<my.app.box>=app indexer etl
HOST_<my.scmandgit.box>=subversion cvs gerrit
```

12. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -V
```

Do this on the SCM server. We'll call this my.scmandgit.box

13. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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. Install the TeamForge, source control and GIT packages.

```
yum install teamforge-scm teamforge-git
```

15. Copy the `site-options.conf` file from the application box to the `/opt/collabnet/teamforge-installer/7.0.0.1/conf` directory on the my.scmandgit.box.
16. Modify the host token settings on the `site-options.conf` file.

```
HOST_localhost=subversion cvs gerrit
DOMAIN_localhost=<scm.domain.com>
HOST_<my.app.box>=app indexer etl
HOST_<my.databaseanddatamart.box>=database datamart
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

17. Create the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

18. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./bootstrap-data.sh
```

19. Start TeamForge.

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

Do this on my.app.box:

20. Set up the initial site data.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./bootstrap-data.sh
```

21. 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.

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

22. Integrate the CLI reports by running the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

23. Run the following initial ETL load jobs.
    a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

    b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

24. Revoke the super user permissions of database and datamart users.

```
/opt/collabnet/teamforge/runtime/scripts/revoke-superuser-permission.py
```

Do this in myscmandgit.box

25. Integrate gerrit by running the `post-install.py` script.

Tip: Skip this step if you are not installing git integration.

```
/opt/collabnet/gerrit/scripts/post-install.py
```

The post installation script will try to detect the required configuration parameters. The following 3 parameters do not have default values and you will be asked to provide them:

- Teamforge Login name: the dedicated TeamForge site administrator account that does not expire and cannot be locked
- Teamforge Password: the password for the above account
- Database password: the password to protect Gerrit's database from unauthorized access. Specify its value when you first run the post-install script. Make sure you note the value because you will be asked for it later.

a) Restart the Gerrit services.

```
/etc/init.d/collabnet restart gerrit
```

b) To verify the Git integration, run the following command.

```
/etc/init.d/collabnet status
```

Do this on my.databaseanddatamart.box

26. Run the following script to set permissions for the TeamForge database read-only user specified by the *DATABASE_READ_ONLY_USER* token.

```
/opt/collabnet/teamforge/runtime/scripts/set-ctfdb-readonly-user-
permission.py
```

27. Run the following script to set permissions for the reporting database read-only user.

```
/opt/collabnet/teamforge/runtime/scripts/set-reports-readonly-user-
permission.py
```

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Install TeamForge the advanced way

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.

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 principle, a multi-box 7.0 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 TeamForge 7.0 with source control on a separate box

In this option, we install the site database, the datamart and the reporting service on the main application box and source control services on a separate box.

Important: There are no changes in the steps for installing TeamForge 7.0 Patch 1

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 is adviced to use port 5632 for the reporting database.

Note: In a multi-box installation of TeamForge, all machines must have the same time zone - without this, ETL will not function properly.

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

1. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.
   - The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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 7 for details.
3. Set the TeamForge repository configuration.

   If you are installing TeamForge with internet access:
   - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
   - Copy it to `/etc/yum.repos.d/`.
   - Refresh your repository cache.

     ```
     yum clean all
     ```

   If you are installing TeamForge without internet access:

   a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

   b) Unpack the disconnected installation package.

   ```
   rpm -ivh <package-name>
   ```

   c) **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

   Insert the Red Hat installation DVD.

   This will provide some of the necessary software and utilities for installing TeamForge.

   d) Mount the Red Hat installation DVD.

   ```
   cd /media/
   mkdir cdrom
   mount /dev/cdrom ./cdrom/
   ```

   Replace `cdrom` with the identifier for your server's CD drive, if necessary.

Tip: If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

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/7.0.0.1
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. Set up your site's master configuration file.

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

At this stage, we'll make the minimal modifications required to set up a usable TeamForge 7.0 site.

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

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) 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
```

c) Configure the database and the datamart.

Tip: For more information about configuring variables, see *site-options.conf* on page 350.

```
DATABASE_TYPE=postgresql
DATABASE_USERNAME=ctfuser
DATABASE_PASSWORD=ctfpwd
DATABASE_NAME=ctfdb
DATABASE_READ_ONLY_USER=ctfrouser
DATABASE_READ_ONLY_PASSWORD=ctfropwd
DATABASE_MAX_POOL_SIZE=220
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_MAX_POOL_SIZE=30
```

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

d)

Important: If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

e) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

f) **It is mandatory to include the *SCM_DEFAULT_SHARED_SECRET* token in the `site-options.conf` file of the primary TeamForge server, and provide it with a value of 16-24 characters. Remember to use the same key in the external SCM integration server also.**

g) Save the file after you make the necessary changes in the `site-options.conf` file.

7. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./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    <DATABASE_NAME>                 <DATABASE_READ_ONLY_USERNAME>
<IP address of my.app.box>/32   md5
```

```
exit
```

c) Create the database user and password that is used to connect from TeamForge.

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

d) Create the database read-only user that is used to connect from TeamForge.

```
createuser -P -S --no-createdb --no-createrole
<database_readonlyusername>
```

e) Create the database.

```
createdb -E UTF8 -O <ctfdatabase_username> <ctfdatabasename>
```

f) Restart PostgreSQL.

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

9. Set up the reporting database (datamart).

Note: For small dataset customers, use the same instance as that of TeamForge database. Do not specify the *REPORTS_DATABASE_PORT*

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 database box>'
```

```
port=5632
```

b) Configure the datamart access.

```
vi /var/lib/pgsql/9.0/reports/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    <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) Create the database user and password that is used to connect from TeamForge.

```
su - postgres
createuser -p <reports_database_port> -P -S --createdb --no-createrole
<reports_database_username>
```

d) Create the database read-only user that is used to connect from TeamForge.

```
createuser -p <reports_database_port> -P -S --no-createdb --no-
createrole <reports_database_readonlyusername>
```

e) Create the database.

```
createdb -p <reports_database_port> -E UTF8 -O <reports database
username> <reports_database_name>
```

f) Restart PostgreSQL.

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

10. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./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
```

13. Run the following script to set permissions for the TeamForge database read only user specified by the *DATABASE_READ_ONLY_USER* token.

```
/opt/collabnet/teamforge/runtime/scripts/set-ctfdb-readonly-user-
permission.py
```

14. Run the following script after bootstrap or migration.

```
/opt/collabnet/teamforge/runtime/scripts/set-reports-readonly-user-
permission.py
```

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

15. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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 7 for details.

17. Set the TeamForge repository configuration.

If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
yum clean all
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

Insert the Red Hat installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) Mount the Red Hat installation DVD.

```
cd /media/
mkdir cdrom
mount /dev/cdrom ./cdrom/
```

Replace `cdrom` with the identifier for your server's CD drive, if necessary.

Tip: If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

18. 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
```

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

d) Save the file after you make the necessary changes in the `site-options.conf` file.

20. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

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

```
export http_proxy=
```

22. 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
```

23. Start TeamForge.

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

Do this on `my.app.box`:

24. 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.

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.

- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

25. Integrate the CLI reports by running the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

26. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

Go to *Verify your TeamForge 7.0 installation* for recommended steps to get started with your site.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Option 2: Distribute TeamForge 7.0 services using Oracle

In this option, we will use 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.

Note: In a multi-box installation of TeamForge, all machines must have the same time zone - without this, ETL will not function properly.

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

1. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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 7 for details.

3. Set the TeamForge repository configuration.

If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
yum clean all
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) 👉 **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

Insert the Red Hat installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) Mount the Red Hat installation DVD.

```
cd /media/
mkdir cdrom
mount /dev/cdrom ./cdrom/
```

Replace `cdrom` with the identifier for your server's CD drive, if necessary.

👉 **Tip:** If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

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/7.0.0.1
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. Set up your site's master configuration file.

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

At this stage, we'll make the minimal modifications required to set up a usable TeamForge 7.0 site.

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

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) 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
```

c) Configure the database and the datamart.

Tip: For more information about configuring variables, see *site-options.conf* on page 350.

```
DATABASE_TYPE=oracle
DATABASE_USERNAME=sitedatabaseusername
DATABASE_PASSWORD=sitedatabasepwd
DATABASE_NAME=sitedatabaseinstancename
DATABASE_MAX_POOL_SIZE=220
REPORTS_DATABASE_USERNAME=reportingdatabaseusername
REPORTS_DATABASE_PASSWORD=reportingdatabasepwd
REPORTS_DATABASE_NAME=reportingdatabaseinstancename
REPORTS_DATABASE_READ_ONLY_USER=reportingreadonlyusername
REPORTS_DATABASE_READ_ONLY_PASSWORD=reportingreadonlyuserpwd
REPORTS_DATABASE_MAX_POOL_SIZE=30
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`.

d) **Important:** If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

e) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

f) **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.

g) 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. Create the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./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 / CentOS 6.3/6.4 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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 11G (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. Copy the Oracle datamart setup script..

```
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 238 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 (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./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
```

18. Run the following script to set permissions for the TeamForge database read only user specified by the *DATABASE_READ_ONLY_USER* token.

```
/opt/collabnet/teamforge/runtime/scripts/set-ctfdb-readonly-user-
permission.py
```

19. Run the following script to set permissions for the TeamForge reports database read only user specified by the *REPORTS_DATABASE_READ_ONLY_USER* token.

```
/opt/collabnet/teamforge/runtime/scripts/set-reports-readonly-user-
permission.py
```

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

20. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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.

21. Check your basic networking setup.

    See *Set up networking for your TeamForge box* on page 7 for details.

22. Set the TeamForge repository configuration.

    If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
yum clean all
```

    If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) 👉 **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

Insert the Red Hat installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) Mount the Red Hat installation DVD.

```
cd /media/
mkdir cdrom
mount /dev/cdrom ./cdrom/
```

Replace `cdrom` with the identifier for your server's CD drive, if necessary.

👉 **Tip:** If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

23. 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
   ```

24. 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) Don't forget to save the file.

25. Create the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

26. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./bootstrap-data.sh
```

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

```
export http_proxy=
```

28. 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
```

29. Start TeamForge.

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

Do this on `my.app.box`

30. 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.

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

31. Integrate the CLI reports by running the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

32. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

Go to *Verify your TeamForge 7.0 installation* for recommended steps to get started with your site.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Option 3: Distribute TeamForge 7.0 on three boxes for intensive database utilization

In this option, we install the reporting service on the main application server, both databases on another server, and source control on a third server. This is appropriate when intensive database utilization is a factor.

Important: There are no changes in the steps for installing TeamForge 7.0 Patch 1

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. It is advised to use port 5632 for the reporting database if there is heavy traffic.

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.

Note: In a multi-box installation of TeamForge, all machines must have the same time zone - without this, ETL will not function properly.

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

1. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.
   - The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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 7 for details.
3. Set the TeamForge repository configuration.

   If you are installing TeamForge with internet access:

   - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
   - Copy it to `/etc/yum.repos.d/`.
   - Refresh your repository cache.

     ```
     yum clean all
     ```

   If you are installing TeamForge without internet access:

   a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

Insert the Red Hat installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) Mount the Red Hat installation DVD.

```
cd /media/
mkdir cdrom
mount /dev/cdrom ./cdrom/
```

Replace `cdrom` with the identifier for your server's CD drive, if necessary.

Tip: If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

4. Install the TeamForge application and reporting service.

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

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

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
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. Set up your site's master configuration file.

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

At this stage, we'll make the minimal modifications required to set up a usable TeamForge 7.0 site.

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

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) Identify the servers and the services running on them.

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

c) Configure the database and the datamart.

Tip: For more information about configuring variables, see *site-options.conf* on page 350.

```
DATABASE_TYPE=postgresql
DATABASE_USERNAME=ctfuser
DATABASE_PASSWORD=ctfpwd
DATABASE_NAME=ctfdb
DATABASE_READ_ONLY_USER=ctfrouser
DATABASE_READ_ONLY_PASSWORD=ctfropwd
DATABASE_MAX_POOL_SIZE=220
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
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

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

We also recommend the following:

- For small dataset customers: use the same instance as that of `ctfdb`; do not specify REPORTS_DATABASE_PORT.
- For medium to large dataset customers: use a separate instance; specify REPORTS_DATABASE_PORT.

d) **Important:** If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

e) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

f) **It is mandatory to include the *SCM_DEFAULT_SHARED_SECRET* token in the `site-options.conf` file of the primary TeamForge server, and provide it with a value of 16-24 characters. Remember to use the same key in the external SCM integration server also.**

g) Save the file after you make the necessary changes in the `site-options.conf` file.

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

```
export http_proxy=
```

8. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./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 / CentOS 6.3/6.4 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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 9.0.13.

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" 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    <DATABASE_NAME>             <DATABASE_READ_ONLY_USERNAME>
<IP address of my.app.box>/32   md5
```

```
exit
```

c) Create the database user and password that is used to connect from TeamForge.

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

d) Create the database read-only user that is used to connect from TeamForge.

```
createuser -P -S --no-createdb --no-createrole
<database_readonlyusername>
```

e) Create the database.

```
createdb -E UTF8 -O <ctfdatabase_username> <ctfdatabasename>
```

f) Restart PostgreSQL.

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

12. Set up the reporting database (datamart).

Note: For small dataset customers, use the same instance as that of TeamForge database. Do not specify the *REPORTS_DATABASE_PORT*

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 database box>'
```

```
port=5632
```

b) Configure the datamart access.

```
vi /var/lib/pgsql/9.0/reports/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    <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) Create the database user and password that is used to connect from TeamForge.

```
su - postgres
createuser -p <reports_database_port> -P -S --createdb --no-createrole
<reports_database_username>
```

d) Create the database read-only user that is used to connect from TeamForge.

```
createuser -p <reports_database_port> -P -S --no-createdb --no-
createrole <reports_database_readonlyusername>
```

e) Create the database.

```
createdb -p <reports_database_port> -E UTF8 -O <reports database
username> <reports_database_name>
```

f) Restart PostgreSQL.

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

Do this on `my.app.box`

13. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./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. Run the following script to set permissions for the TeamForge database read only user specified by the *DATABASE_READ_ONLY_USER* token.

```
/opt/collabnet/teamforge/runtime/scripts/set-ctfdb-readonly-user-
permission.py
```

16. Run the following script after bootstrap or migration.

```
/opt/collabnet/teamforge/runtime/scripts/set-reports-readonly-user-
permission.py
```

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

17. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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.

18. Check your basic networking setup.

See *Set up networking for your TeamForge box* on page 7 for details.

19. Set the TeamForge repository configuration.

If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
yum clean all
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c)

Note: If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

Insert the Red Hat installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) Mount the Red Hat installation DVD.

```
cd /media/
mkdir cdrom
```

```
mount /dev/cdrom ./cdrom/
```

Replace `cdrom` with the identifier for your server's CD drive, if necessary.

Tip: If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

20. Install the TeamForge source control service.

```
yum install teamforge-scm
```

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

a) Identify the servers 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) Don't forget to save the file.

22. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

23. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./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
```

27. Integrate the CLI reports by running the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

28. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Verify your TeamForge 7.0 installation

Congratulations: you have just installed your TeamForge 7.0 site. 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 246 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 login to TeamForge application.

3. To install your license key.

   See *Supply your TeamForge license key from Teamforge user interface* on page 234.

4. To verify that the tracker initial load (triggered through `TrackerInitialJob`) completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

   ```
   select * from etl_job where job_name ='tracker_initial_etl'
   ```

   The status column should have a value of 1.

5. To create a sample project.

   See *Create a TeamForge project*.

6. To 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 TeamForge 7.0 site* on page 232 to help keep your site going.

Uninstall TeamForge 7.0

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.

Install CollabNet TeamForge 7.0 on SuSE

Install TeamForge 7.0 with all services on the same box

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: There are no changes in the steps for installing TeamForge 7.0 Patch 1

In TeamForge 7.0, this scenario includes installing Black Duck Code Sight and Git Integration as well on this box.

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 SuSE Linux Enterprise Server 11 SP2 and log in as root.
   - See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
   - See *the SuSE Linux Enterprise Server deployment 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 7 for details.

3. Set up the SuSE repository.

   a) Register the machine with Novell.

      ```
      suse_register -i -a email=<MailID> -a regcode-sles=<Registration code>
      ```

      where `<MailID>` is the email address associated with your SuSE Linux Enterprise Server license and `<Registration code>` is the Suse Enterprise Linux license code.

   b) Move the existing installer repository out of the way.

      ```
      cd /etc/zypp/repos.d/
      mv SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo_old
      ```

   c) Make sure the correct Novell repository is enabled.

      Open `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo` in a text editor and confirm that `enabled=1`.

   d) Refresh `zypper`.

      ```
      zypper ref
      ```

      Note: As an alternative, you can insert the SuSe Linux Enterprise Server DVD.

      - Ensure that the DVD is mounted in the location specified by the *baseurl* variable in `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo`.
      - Use `zypper info postgresql` to check that the DVD is properly mounted.

4. If you are installing TeamForge without internet access, contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

a) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

b) **Note:** If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

Insert the SuSE installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

c) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

If necessary, replace `sr0` with the identifier for your server's CD drive.

Now the package manager will look for the SuSE installation components it needs on the DVD and not on the Internet.

5. Install the following application packages.

a) TeamForge: To install the TeamForge application packages run the following command:

```
zypper install teamforge
```

b) GIT: To install the GIT packages run the following command:

```
zypper install teamforge-git
```

c) Black Duck Code Sight: To install Black Duck Code Sight run the following command

```
zypper install teamforge-codesearch
```

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

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

At this stage, we'll make the minimal modifications required to set up a usable TeamForge 7.0 site.

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

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 site-wide reporting functionality by adding the *REPORTS_** variables.

See *Turn on site-wide reporting* on page 250 for details.

d) By default, the site wide reports are not commented. To turn on the site wide reports, set the values. Else, comment out the following reports.

- REPORTS_DATABASE_NAME=
- REPORTS_DATABASE_PASSWORD=
- REPORTS_DATABASE_USERNAME=
- REPORTS_DATABASE_READ_ONLY_USER=
- REPORTS_DATABASE_READ_ONLY_PASSWORD=

e) **It is mandatory to include the *SCM_DEFAULT_SHARED_SECRET* token in the `site-options.conf` file of the primary TeamForge server, and provide it with a value of 16-24 characters. Remember to use the same key in the external SCM integration server also.**

f) If the token *REQUIRE_RANDOM_ADMIN_PASSWORD* is already set to true, then set the token *ADMIN_EMAIL* with a valid email address.

```
ADMIN_EMAIL=root@{__APPLICATION_HOST__}
```

g) 👉 **Important:** If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

h) Include the list of TeamForge user accounts whose password should not expire.

```
USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer
```

If you are installing git integration, add a TeamForge user for example,"gitadmin" with site-administrator rights and append the username against this parameter.

```
USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer,gitadmin
```

i) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

j) Configure the following settings for Git:

```
HOST_my.host.name=app etl database datamart subversion cvs
 indexer gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

k) Configure the following settings for Black Duck Code Sight.

```
HOST_localhost=app etl database datamart subversion cvs
 indexer codesearch
```

👉 **Note:** In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>

To enable SSL for Black Duck Code Sight, include this token:
BDCS_SSL=on
```

👉 **Note:** If you have Black Duck Code Sight on SSL and the following external certificate tokens are not provided, the installer will generate a self-signed certificate.

```
For valid SSL certificates, configure the following tokens:
BDCS_SSL_CERT_FILE=
BDCS_SSL_KEY_FILE=
The ca.crt and chain files are optional -- leave out the tokens if you
 don't use the files.
BDCS_SSL_CA_CERT_FILE=
```

```
BDCS_SSL_CHAIN_FILE=
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
BDCS_ADMIN_PASSWORD=<password>
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

Advanced Black Duck Code Sight configuration settings:

Note: The following are the advanced configuration tokens which can be set once during the time of installation.

```
The path where the repositories enabled for codesearch to check out.
BDCS_SCAN_SOURCE_DIR_ROOT=/opt/collabnet/blackduck/scan

The path where the codesearch software is installed.
BDCS_INSTALL_PATH=/opt/collabnet/blackduck

The path where codesearch database is installed.
BDCS_PGSQL_HOME_DIR_ROOT=/opt/collabnet/blackduck/postgres

The port number for codesearch db server.
BDCS_PGSQL_PORT=55435

The tomcat maximum heap memory size in megabytes.
BDCS_TOMCAT_MX_IN_MB=1024

The shutdown port number for codesearch tomcat server.
BDCS_TOMCAT_SHUTDOWN_PORT=9189
```

7. Create the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

8. Set up the initial site data (bootstrap).

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

9. Start TeamForge.

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

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

10. If you have installed Black Duck Code Sight, then install the license for Black Duck Code Sight. For more information, see *Install the Black Duck Code Sight license* on page 274.
11. Integrate gerrit by running the `post-install.py` script.

Tip: Skip this step if you are not installing git integration.

```
/opt/collabnet/gerrit/scripts/post-install.py
```

The post installation script will try to detect the required configuration parameters. The following 3 parameters do not have default values and you will be asked to provide them:

- Teamforge Login name: the dedicated TeamForge site administrator account that does not expire and cannot be locked
- Teamforge Password: the password for the above account
- Database password: the password to protect Gerrit's database from unauthorized access. Specify its value when you first run the post-install script. Make sure you note the value because you will be asked for it later.

a) Restart the Gerrit services.

```
/etc/init.d/collabnet restart gerrit
```

b) To verify the Git integration, run the following command.

```
/etc/init.d/collabnet status
```

12. Integrate the CLI reports by running the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

13. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

14. Integrate Black Duck Code Sight with TeamForge by running the Black Duck Code Sight `post-install.sh` script.

Note: It is assumed that Subversion's client configuration file (`/root/.subversion/config`) for the root user is the default one without customization.

Note: If the Black Duck Code Sight is running on a separate server, run the following command in the code sight server.

```
sudo /opt/collabnet/teamforge/runtime/scripts/codesearch/post-install.sh
```

15. After you run the Black Duck Code Sight `post-install.sh` script, run the following script from the same directory:

```
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL similar to "http://myint.box.net/svn/repos" where `myint.box` is the box with the SCM integration server.

Also, if you add a new integration server at some point later, you'll need to run this script, (after creating new integration server), in the TeamForge `app` box.

Important: Do this if you have a multi-box setup with a SCM integration server on a separate server.

16. If the token *VALIDATE_SSL_CERTS* is set to "true", you must run the codesearch runtime script *trust-cert.sh* in the application box.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
```

17. Restart the `collabnet` services.

```
/etc/init.d/collabnet restart
```

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

a) 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.

b) Create a sample project.

See *Create a TeamForge project*.

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

See *Create a site-wide broadcast*.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Install TeamForge 7.0 with database and datamart on a separate box

In this option, we install database and datamart (reporting database) on a separate box and other services on the main application box.

Important: There are no changes in the steps for installing TeamForge 7.0 Patch 1

We call this a "dedicated install" since both the servers are dedicated to TeamForge. TheTeamForge installer takes care of the database configurations. 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. It is advised to use port 5632 for the reporting database if there is heavy traffic.

Important: If the remote machine where the datamart is running is not under your direct control, check with the database administrator to make sure that you can carry out these instructions on that box.

Note: In a multi-box installation of TeamForge, ensure that all servers have the same system time zone for ETL to function properly.

Do this on my.app.box

1. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
- See *the SuSE Linux Enterprise Server deployment 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 7 for details.

3. Set up the SuSE repository.

a) Register the machine with Novell.

```
suse_register -i -a email=<MailID> -a regcode-sles=<Registration code>
```

where `<MailID>` is the email address associated with your SuSE Linux Enterprise Server license and `<Registration code>` is the Suse Enterprise Linux license code.

b) Move the existing installer repository out of the way.

```
cd /etc/zypp/repos.d/
mv SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo SUSE-Linux-
Enterprise-Server-11-SP1\ 11.1.1-1.152.repo_old
```

c) Make sure the correct Novell repository is enabled.

Open `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo` in a text editor and confirm that `enabled=1`.

d) Refresh `zypper`.

```
zypper ref
```

Note: As an alternative, you can insert the SuSe Linux Enterprise Server DVD.

- Ensure that the DVD is mounted in the location specified by the *baseurl* variable in `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo`.
- Use `zypper info postgresql` to check that the DVD is properly mounted.

4. If you are installing TeamForge with internet access:

   - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
   - Copy it to `/etc/yum.repos.d/`.
   - Refresh your repository cache.

     ```
     zypper ref
     ```

   If you are installing TeamForge without internet access:

   a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

   b) Unpack the disconnected installation package.

   ```
   rpm -ivh <package-name>
   ```

   c)

   Note: If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

   Insert the SuSE installation DVD.

   This will provide some of the necessary software and utilities for installing TeamForge.

   d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

   If necessary, replace `sr0` with the identifier for your server's CD drive.

   Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.

5. Install the following TeamForge application, ETL and source control services.

   ```
   zypper install teamforge-app teamforge-etl teamforge-scm
   ```

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

   ```
   vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
   ```

   At this stage, we'll make the minimal modifications required to set up a usable TeamForge 7.0 site.

   Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

   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) Identify the boxes and the services running on them.

```
HOST_localhost=app etl subversion cvs indexer
DOMAIN_localhost=worlddomination.supervillain.org
HOST_<my.databaseanddatamart.box>=datamart
```

c) Configure the database and datamart.

Tip: For more information about configuring variables, see *site-options.conf* on page 350.

```
DATABASE_TYPE=postgresql
DATABASE_USERNAME=ctfuser
DATABASE_PASSWORD=ctfpwd
DATABASE_NAME=ctfdb
DATABASE_MAX_POOL_SIZE=220
DATABASE_READ_ONLY_USER=ctfrouser
DATABASE_READ_ONLY_PASSWORD=ctfropwd
REPORTS_DATABASE_USERNAME=ctfrptuser
REPORTS_DATABASE_PASSWORD=ctfrptpwd
REPORTS_DATABASE_NAME=ctfrptdb
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

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

We also recommend the following:

- For small dataset customers: use the same instance as that of `ctfdb`; do not specify REPORTS_DATABASE_PORT.
- For medium to large dataset customers: use a separate instance; specify REPORTS_DATABASE_PORT.

d) **Important:** If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

e) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

f) **It is mandatory to include the *SCM_DEFAULT_SHARED_SECRET* token in the `site-options.conf` file of the primary TeamForge server, and provide it with a value of 16-24 characters. Remember to use the same key in the external SCM integration server also.**

7. Create the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

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

```
export http_proxy=
```

Do this on the database or datamart server. We'll call this my.databaseanddatamart.box

9. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
- See *the SuSE Linux Enterprise Server deployment guide* for help.

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

10. Install the TeamForge database packages.

```
zypper install teamforge-database
```

11. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

Do this on my.app.box

12. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./bootstrap-data.sh
```

13. 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.

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

14. Integrate the CLI reports by running the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

15. Run the following initial ETL load jobs.
    a) Change to the `runtime/scripts` directory.

    ```
    cd /opt/collabnet/teamforge/runtime/scripts
    ```

    b) Run the `TrackerInitialJob`.

    ```
    ./etl-client.py -r TrackerInitialJob
    ```

    c) Run the `SCMInitialJob`.

    ```
    ./etl-client.py -r SCMCommitInitialJob
    ```

    Tip: For more information see *When do I run the initial load job?* on page 294.

16. Revoke the super user permissions of database and datamart users.

```
/opt/collabnet/teamforge/runtime/scripts/revoke-superuser-permission.py
```

Do this in my.databaseanddatamart.box

17. Run the following script to set permissions for the TeamForge database read-only user specified by the *DATABASE_READ_ONLY_USER* token.

```
/opt/collabnet/teamforge/runtime/scripts/set-ctfdb-readonly-user-
permission.py
```

18. Run the following script to set permissions for the reporting database read-only user.

```
/opt/collabnet/teamforge/runtime/scripts/set-reports-readonly-user-
permission.py
```

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Install TeamForge 7.0 with reporting services on a separate box

In this option, we install the ETL and datamart (reporting database) on a separate box and other services on the main application box.

Important: There are no changes in the steps for installing TeamForge 7.0 Patch 1

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

1. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.
   - See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
   - See *the SuSE Linux Enterprise Server deployment 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 7 for details.
3. Set up the SuSE repository.
   a) Register the machine with Novell.

   ```
   suse_register -i -a email=<MailID> -a regcode-sles=<Registration code>
   ```

   where `<MailID>` is the email address associated with your SuSE Linux Enterprise Server license and `<Registration code>` is the Suse Enterprise Linux license code.
   b) Move the existing installer repository out of the way.

   ```
   cd /etc/zypp/repos.d/
   mv SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo_old
   ```

   c) Make sure the correct Novell repository is enabled.

   Open `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo` in a text editor and confirm that `enabled=1`.
   d) Refresh `zypper`.

   ```
   zypper ref
   ```

   Note: As an alternative, you can insert the SuSe Linux Enterprise Server DVD.
   - Ensure that the DVD is mounted in the location specified by the *baseurl* variable in `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo`.
   - Use `zypper info postgresql` to check that the DVD is properly mounted.
4. If you are installing TeamForge with internet access:
   - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
   - Copy it to `/etc/yum.repos.d/`.
   - Refresh your repository cache.

   ```
   zypper ref
   ```

   If you are installing TeamForge without internet access:
   a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) **Note:** If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

Insert the SuSE installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

If necessary, replace `sr0` with the identifier for your server's CD drive.

Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.

5. Install the following TeamForge application, database and other source control services.

```
zypper install teamforge-app teamforge-database teamforge-scm
```

6. Install the following application packages.

a) GIT: To install the GIT packages run the following command.

```
zypper install teamforge-git
```

b) To install Black Duck Code Sight run the following command.

```
zypper install teamforge-codesearch
```

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

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

At this stage, we'll make the minimal modifications required to set up a usable TeamForge 7.0 site.

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

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) Identify the boxes and the services running on them.

```
HOST_localhost=app database subversion cvs
DOMAIN_localhost=worlddomination.supervillain.org
HOST_<my.datamartetlbox>=datamart etl
```

c) Review the *DATABASE* variables.

You may want to substitute your own values for the default database names, user names and passwords.

d) Turn on the site-wide reporting functionality by adding the *REPORTS_** variables.

See *Turn on site-wide reporting* on page 250 for details.

e) By default, the site wide reports are not commented. To turn on the site wide reports, set the values. Else, comment out the following reports.

- REPORTS_DATABASE_NAME=
- REPORTS_DATABASE_PASSWORD=
- REPORTS_DATABASE_USERNAME=
- REPORTS_DATABASE_READ_ONLY_USER=
- REPORTS_DATABASE_READ_ONLY_PASSWORD=

f) **It is mandatory to include the *SCM_DEFAULT_SHARED_SECRET* token in the `site-options.conf` file of the primary TeamForge server, and provide it with a value of 16-24 characters. Remember to use the same key in the external SCM integration server also.**

g) If the token *REQUIRE_RANDOM_ADMIN_PASSWORD* is already set to true, then set the token *ADMIN_EMAIL* with a valid email address.

```
ADMIN_EMAIL=root@{__APPLICATION_HOST__}
```

h)

Important: If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

i) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

j) Configure the following settings for Git.

```
HOST_my.host.name=app etl database datamart subversion cvs
 indexer gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

k) Configure the following settings for Black Duck Code Sight.

```
HOST_localhost=app etl database datamart subversion cvs
 indexer codesearch
```

Note: In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>

To enable SSL for Black Duck Code Sight, include this token:
BDCS_SSL=on
```

Note: If you have Black Duck Code Sight on SSL and the following external certificate tokens are not provided, the installer will generate a self-signed certificate.

```
For valid SSL certificates, configure the following tokens:
BDCS_SSL_CERT_FILE=
BDCS_SSL_KEY_FILE=
The ca.crt and chain files are optional -- leave out the tokens if you
 don't use the files.
BDCS_SSL_CA_CERT_FILE=
BDCS_SSL_CHAIN_FILE=
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
BDCS_ADMIN_PASSWORD=<password>
```

```
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

Advanced Black Duck Code Sight configuration settings:

Note: The following are the advanced configuration tokens which can be set once during the time of installation.

```
The path where the repositories enabled for codesearch to check out.
BDCS_SCAN_SOURCE_DIR_ROOT=/opt/collabnet/blackduck/scan

The path where the codesearch software is installed.
BDCS_INSTALL_PATH=/opt/collabnet/blackduck

The path where codesearch database is installed.
BDCS_PGSQL_HOME_DIR_ROOT=/opt/collabnet/blackduck/postgres

The port number for codesearch db server.
BDCS_PGSQL_PORT=55435

The tomcat maximum heap memory size in megabytes.
BDCS_TOMCAT_MX_IN_MB=1024

The shutdown port number for codesearch tomcat server.
BDCS_TOMCAT_SHUTDOWN_PORT=9189
```

8. Create the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

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 database or datamart server. We'll call this my.databaseanddatamart.box

10. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
- See *the SuSE Linux Enterprise Server deployment guide* for help.

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

11. Install the TeamForge database and ETL packages.

```
zypper install teamforge-database teamforge-etl
```

12. Copy the `site-options.conf` from the application box.

13. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

Do this on my.app.box.

14. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./bootstrap-data.sh
```

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.

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

16. If you have installed Black Duck Code Sight, then install the license for Black Duck Code Sight. For more information, see *Install the Black Duck Code Sight license* on page 274.
17. Integrate gerrit by running the `post-install.py` script.

    Tip: Skip this step if you are not installing git integration.

    ```
    /opt/collabnet/gerrit/scripts/post-install.py
    ```

    The post installation script will try to detect the required configuration parameters. The following 3 parameters do not have default values and you will be asked to provide them:

    - Teamforge Login name: the dedicated TeamForge site administrator account that does not expire and cannot be locked
    - Teamforge Password: the password for the above account
    - Database password: the password to protect Gerrit's database from unauthorized access. Specify its value when you first run the post-install script. Make sure you note the value because you will be asked for it later.

    a) Restart the Gerrit services.

    ```
    /etc/init.d/collabnet restart gerrit
    ```

    b) To verify the Git integration, run the following command.

    ```
    /etc/init.d/collabnet status
    ```

18. Integrate the CLI reports by running the `post-install.py` script.

    ```
    sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
    ```

19. Run the following initial ETL load jobs.

    a) Change to the `runtime/scripts` directory.

    ```
    cd /opt/collabnet/teamforge/runtime/scripts
    ```

    b) Run the `TrackerInitialJob`.

    ```
    ./etl-client.py -r TrackerInitialJob
    ```

    c) Run the `SCMInitialJob`.

    ```
    ./etl-client.py -r SCMCommitInitialJob
    ```

    Tip: For more information see *When do I run the initial load job?* on page 294.

20. Integrate Black Duck Code Sight with TeamForge by running the Black Duck Code Sight `post-install.sh` script.

Note: It is assumed that Subversion's client configuration file (`/root/.subversion/config`) for the root user is the default one without customization.

Note: If the Black Duck Code Sight is running on a separate server, run the following command in the code sight server.

```
sudo /opt/collabnet/teamforge/runtime/scripts/codesearch/post-install.sh
```

21. After you run the Black Duck Code Sight `post-install.sh` script, run the following script from the same directory:

    ```
    ./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
    ```

    Provide a repository base URL similar to "http://myint.box.net/svn/repos" where `myint.box` is the box with the SCM integration server.

    Also, if you add a new integration server at some point later, you'll need to run this script, (after creating new integration server), in the TeamForge `app` box.

    Important: Do this if you have a multi-box setup with a SCM integration server on a separate server.

22. If the token *VALIDATE_SSL_CERTS* is set to "true", you must run the codesearch runtime script *trust-cert.sh* in the application box.

    ```
    cd /opt/collabnet/teamforge/runtime/scripts/codesearch
    ./trust-cert.sh
    ```

23. Restart the `collabnet` services.

    ```
    /etc/init.d/collabnet restart
    ```

24. Apply some finishing touches and make sure everything is running smoothly.
    a) 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.
    b) Create a sample project.

       See *Create a TeamForge project*.
    c) Write a welcome message to your site's users.

       See *Create a site-wide broadcast*.

25. Revoke the super user permissions of database and datamart users.

    ```
    /opt/collabnet/teamforge/runtime/scripts/revoke-superuser-permission.py
    ```

26. Run the following script to set permissions for the reporting database read-only user.

    ```
    /opt/collabnet/teamforge/runtime/scripts/set-reports-readonly-user-permission.py
    ```

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Install Black Duck Code Sight on SuSE

In this option, we install Black Duck Code Sight on a separate box on SuSE and other services on the main application box.

Important: There are no changes in the steps for installing TeamForge 7.0 Patch 1

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

1. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.
   - See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
   - See *the SuSE Linux Enterprise Server deployment 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 7 for details.

3. Set up the SuSE repository.

   a) Register the machine with Novell.

   ```
   suse_register -i -a email=<MailID> -a regcode-sles=<Registration code>
   ```

   where `<MailID>` is the email address associated with your SuSE Linux Enterprise Server license and `<Registration code>` is the Suse Enterprise Linux license code.

   b) Move the existing installer repository out of the way.

   ```
   cd /etc/zypp/repos.d/
   mv SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo_old
   ```

   c) Make sure the correct Novell repository is enabled.

   Open `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo` in a text editor and confirm that `enabled=1`.

   d) Refresh `zypper`.

   ```
   zypper ref
   ```

   Note: As an alternative, you can insert the SuSe Linux Enterprise Server DVD.

   - Ensure that the DVD is mounted in the location specified by the *baseurl* variable in `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo`.
   - Use `zypper info postgresql` to check that the DVD is properly mounted.

4. If you are installing TeamForge with internet access:

   - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
   - Copy it to `/etc/yum.repos.d/`.
   - Refresh your repository cache.

   ```
   zypper ref
   ```

   If you are installing TeamForge without internet access:

   a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

   b) Unpack the disconnected installation package.

   ```
   rpm -ivh <package-name>
   ```

   c) **Note:** If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

   Insert the SuSE installation DVD.

   This will provide some of the necessary software and utilities for installing TeamForge.

   d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

If necessary, replace `sr0` with the identifier for your server's CD drive.

Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.

5. Install the following TeamForge application, database and other source control services.

```
zypper install teamforge-app teamforge-database teamforge-scm
```

6. Install the following application packages.
   a) GIT: To install the GIT packages run the following command.

```
zypper install teamforge-git
```

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

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

At this stage, we'll make the minimal modifications required to set up a usable TeamForge 7.0 site.

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

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) Identify the boxes and the services running on them.

```
HOST_localhost=app database subversion cvs
DOMAIN_localhost=worlddomination.supervillain.org
HOST_<my.datamartetlbox>=datamart etl
```

c) Review the *DATABASE* variables.

You may want to substitute your own values for the default database names, user names and passwords.

d) Turn on the site-wide reporting functionality by adding the *REPORTS_** variables.

See *Turn on site-wide reporting* on page 250 for details.

e) By default, the site wide reports are not commented. To turn on the site wide reports, set the values. Else, comment out the following reports.

- REPORTS_DATABASE_NAME=
- REPORTS_DATABASE_PASSWORD=
- REPORTS_DATABASE_USERNAME=
- REPORTS_DATABASE_READ_ONLY_USER=
- REPORTS_DATABASE_READ_ONLY_PASSWORD=

f) **It is mandatory to include the *SCM_DEFAULT_SHARED_SECRET* token in the `site-options.conf` file of the primary TeamForge server, and provide it with a value of 16-24 characters. Remember to use the same key in the external SCM integration server also.**

g) If the token *REQUIRE_RANDOM_ADMIN_PASSWORD* is already set to true, then set the token *ADMIN_EMAIL* with a valid email address.

```
ADMIN_EMAIL=root@{__APPLICATION_HOST__}
```

h)

Important: If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

i) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

j) Configure the following settings for Git:

```
HOST_my.host.name=app etl database datamart subversion cvs
 indexer gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

k) Configure the following settings for Black Duck Code Sight.

```
HOST_localhost=app etl database datamart subversion cvs
 indexer codesearch
```

Note: In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>

To enable SSL for Black Duck Code Sight, include this token:
BDCS_SSL=on
```

Note: If you have Black Duck Code Sight on SSL and the following external certificate tokens are not provided, the installer will generate a self-signed certificate.

```
For valid SSL certificates, configure the following tokens:
BDCS_SSL_CERT_FILE=
BDCS_SSL_KEY_FILE=
The ca.crt and chain files are optional -- leave out the tokens if you
 don't use the files.
BDCS_SSL_CA_CERT_FILE=
BDCS_SSL_CHAIN_FILE=
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
BDCS_ADMIN_PASSWORD=<password>
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

Advanced Black Duck Code Sight configuration settings:

Note: The following are the advanced configuration tokens which can be set once during the time of installation.

```


The path where the repositories enabled for codesearch to check out.
BDCS_SCAN_SOURCE_DIR_ROOT=/opt/collabnet/blackduck/scan

The path where the codesearch software is installed.
BDCS_INSTALL_PATH=/opt/collabnet/blackduck

The path where codesearch database is installed.
```

```
BDCS_PGSQL_HOME_DIR_ROOT=/opt/collabnet/blackduck/postgres

The port number for codesearch db server.
BDCS_PGSQL_PORT=55435

The tomcat maximum heap memory size in megabytes.
BDCS_TOMCAT_MX_IN_MB=1024

The shutdown port number for codesearch tomcat server.
BDCS_TOMCAT_SHUTDOWN_PORT=9189
```

8. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

Note: If you are setting up CollabNet TeamForge on a machine that is meant for emergency backup or disaster recovery, use the -R option in place of the -r option.

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 Code Sight server. We'll call this my.codesight.box

10. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
- See *the SuSE Linux Enterprise Server deployment guide* for help.

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

11. Run the following command to install Black Duck Code Sight.

```
zypper install teamforge-codesearch
```

12. Copy the `site-options.conf` from the application box.

13. Run the installer.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

Do this on my.app.box:

14. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./bootstrap-data.sh
```

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.

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.

- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

16. If you have installed Black Duck Code Sight, then install the license for Black Duck Code Sight. For more information, see *Install the Black Duck Code Sight license* on page 274.
17. If yo have installed Gerrit integration, integrate gerrit by running the `post-install.py` script.

    ```
    /opt/collabnet/gerrit/scripts/post-install.py
    ```
18. Integrate the CLI reports by running the `post-install.py` script.

    ```
    sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
    ```
19. Run the following initial ETL load jobs.
    a) Change to the `runtime/scripts` directory.

       ```
       cd /opt/collabnet/teamforge/runtime/scripts
       ```
    b) Run the `TrackerInitialJob`.

       ```
       ./etl-client.py -r TrackerInitialJob
       ```
    c) Run the `SCMInitialJob`.

       ```
       ./etl-client.py -r SCMCommitInitialJob
       ```

    Tip: For more information see *When do I run the initial load job?* on page 294.
20. Integrate Black Duck Code Sight with TeamForge by running the Black Duck Code Sight `post-install.sh` script.

    Note: It is assumed that Subversion's client configuration file (`/root/.subversion/config`) for the root user is the default one without customization.

    Note: If the Black Duck Code Sight is running on a separate server, run the following command in the code sight server.

    ```
    sudo /opt/collabnet/teamforge/runtime/scripts/codesearch/post-install.sh
    ```
21. After you run the Black Duck Code Sight `post-install.sh` script, run the following script from the same directory:

    ```
    ./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
    ```

    Provide a repository base URL similar to "http://myint.box.net/svn/repos" where `myint.box` is the box with the SCM integration server.

    Also, if you add a new integration server at some point later, you'll need to run this script, (after creating new integration server), in the TeamForge `app` box.

    Important: Do this if you have a multi-box setup with a SCM integration server on a separate server.
22. If the token *VALIDATE_SSL_CERTS* is set to "true", you must run the codesearch runtime script *trust-cert.sh* in the application box.

    ```
    cd /opt/collabnet/teamforge/runtime/scripts/codesearch
    ./trust-cert.sh
    ```
23. Restart the `collabnet` services.

    ```
    /etc/init.d/collabnet restart
    ```
24. Apply some finishing touches and make sure everything is running smoothly.
    a) 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.
    b) Create a sample project.

       See *Create a TeamForge project*.

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

See *Create a site-wide broadcast*.

25. Revoke the super user permissions of database and datamart users.

```
/opt/collabnet/teamforge/runtime/scripts/revoke-superuser-permission.py
```

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Install TeamForge 7.0 with Git integration on a separate box

In this option, we install Git integration on a separate box and other services on the main application box.

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

1. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
- See *the SuSE Linux Enterprise Server deployment 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 7 for details.

3. Set up the SuSE repository.

a) Register the machine with Novell.

```
suse_register -i -a email=<MailID> -a regcode-sles=<Registration code>
```

where `<MailID>` is the email address associated with your SuSE Linux Enterprise Server license and `<Registration code>` is the Suse Enterprise Linux license code.

b) Move the existing installer repository out of the way.

```
cd /etc/zypp/repos.d/
mv SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo SUSE-Linux-
Enterprise-Server-11-SP1\ 11.1.1-1.152.repo_old
```

c) Make sure the correct Novell repository is enabled.

Open `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo` in a text editor and confirm that `enabled=1`.

d) Refresh `zypper`.

```
zypper ref
```

Note: As an alternative, you can insert the SuSe Linux Enterprise Server DVD.

- Ensure that the DVD is mounted in the location specified by the *baseurl* variable in `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo`.
- Use `zypper info postgresql` to check that the DVD is properly mounted.

4. If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
zypper ref
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) **Note:** If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

Insert the SuSE installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

If necessary, replace `sr0` with the identifier for your server's CD drive.

Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.

5. Install the following TeamForge application, database and other source control services.

```
zypper install teamforge-app teamforge-database teamforge-scm
```

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

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

At this stage, we'll make the minimal modifications required to set up a usable TeamForge 7.0 site.

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

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) Identify the boxes and the services running on them.

```
HOST_localhost=app database subversion cvs datamart etl indexer
DOMAIN_localhost=worlddomination.supervillain.org
HOST_<my.git.box>=gerrit
```

c) Review the *DATABASE* variables.

You may want to substitute your own values for the default database names, user names and passwords.

d) Turn on the site-wide reporting functionality by adding the *REPORTS_** variables.

See *Turn on site-wide reporting* on page 250 for details.

e) By default, the site wide reports are not commented. To turn on the site wide reports, set the values. Else, comment out the following reports.

- REPORTS_DATABASE_NAME=

- REPORTS_DATABASE_PASSWORD=
- REPORTS_DATABASE_USERNAME=
- REPORTS_DATABASE_READ_ONLY_USER=
- REPORTS_DATABASE_READ_ONLY_PASSWORD=

f) **It is mandatory to include the *SCM_DEFAULT_SHARED_SECRET* token in the `site-options.conf` file of the primary TeamForge server, and provide it with a value of 16-24 characters. Remember to use the same key in the external SCM integration server also.**

g) Include the list of TeamForge user accounts whose password should not expire.

```
USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer
```

If you are installing git integration, add a TeamForge user for example,"gitadmin" with site-administrator rights and append the username against this parameter.

```
USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer,gitadmin
```

h) If the token *REQUIRE_RANDOM_ADMIN_PASSWORD* is already set to true, then set the token *ADMIN_EMAIL* with a valid email address.

```
ADMIN_EMAIL=root@{__APPLICATION_HOST__}
```

i) 👉 **Important:** If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

j) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

k) Configure the following settings for Black Duck Code Sight.

```
HOST_localhost=app etl database datamart subversion cvs
 indexer codesearch
```

👉 **Note:** In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>

To enable SSL for Black Duck Code Sight, include this token:
BDCS_SSL=on
```

👉 **Note:** If you have Black Duck Code Sight on SSL and the following external certificate tokens are not provided, the installer will generate a self-signed certificate.

```
For valid SSL certificates, configure the following tokens:
BDCS_SSL_CERT_FILE=
BDCS_SSL_KEY_FILE=
The ca.crt and chain files are optional -- leave out the tokens if you
 don't use the files.
BDCS_SSL_CA_CERT_FILE=
BDCS_SSL_CHAIN_FILE=
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
BDCS_ADMIN_PASSWORD=<password>
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
```

```
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

Advanced Black Duck Code Sight configuration settings:

Note: The following are the advanced configuration tokens which can be set once during the time of installation.

```
The path where the repositories enabled for codesearch to check out.
BDCS_SCAN_SOURCE_DIR_ROOT=/opt/collabnet/blackduck/scan

The path where the codesearch software is installed.
BDCS_INSTALL_PATH=/opt/collabnet/blackduck

The path where codesearch database is installed.
BDCS_PGSQL_HOME_DIR_ROOT=/opt/collabnet/blackduck/postgres

The port number for codesearch db server.
BDCS_PGSQL_PORT=55435

The tomcat maximum heap memory size in megabytes.
BDCS_TOMCAT_MX_IN_MB=1024

The shutdown port number for codesearch tomcat server.
BDCS_TOMCAT_SHUTDOWN_PORT=9189
```

7. Create the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

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

```
export http_proxy=
```

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

9. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.
   - See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
   - See *the SuSE Linux Enterprise Server deployment guide* for help.

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

10. Run the following command to install the Git packages.

```
zypper install teamforge-git
```

11. Copy the `site-options.conf` from the application box.
12. Configure the following settings for Git:

```
HOST_my.git=gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

13. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

Do this on my.app.box:

14. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
```

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

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.

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

16. If you have installed Black Duck Code Sight, then install the license for Black Duck Code Sight. For more information, see *Install the Black Duck Code Sight license* on page 274.

17. Integrate the CLI reports by running the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

18. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

19. Integrate Black Duck Code Sight with TeamForge by running the Black Duck Code Sight `post-install.sh` script.

Note: It is assumed that Subversion's client configuration file (`/root/.subversion/config`) for the root user is the default one without customization.

Note: If the Black Duck Code Sight is running on a separate server, run the following command in the code sight server.

```
sudo /opt/collabnet/teamforge/runtime/scripts/codesearch/post-install.sh
```

20. After you run the Black Duck Code Sight `post-install.sh` script, run the following script from the same directory:

```
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL similar to "http://myint.box.net/svn/repos" where `myint.box` is the box with the SCM integration server.

Also, if you add a new integration server at some point later, you'll need to run this script, (after creating new integration server), in the TeamForge `app` box.

Important: Do this if you have a multi-box setup with a SCM integration server on a separate server.

21. If the token *VALIDATE_SSL_CERTS* is set to "true", you must run the codesearch runtime script *trust-cert.sh* in the application box.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
```

22. Restart the `collabnet` services.

```
/etc/init.d/collabnet restart
```

23. Apply some finishing touches and make sure everything is running smoothly.
    a) 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.
    b) Create a sample project.
       See *Create a TeamForge project*.
    c) Write a welcome message to your site's users.
       See *Create a site-wide broadcast*.

Do this in my.git.box

24. Integrate gerrit by running the `post-install.py` script.

Tip: Skip this step if you are not installing git integration.

```
/opt/collabnet/gerrit/scripts/post-install.py
```

The post installation script will try to detect the required configuration parameters. The following 3 parameters do not have default values and you will be asked to provide them:

- Teamforge Login name: the dedicated TeamForge site administrator account that does not expire and cannot be locked
- Teamforge Password: the password for the above account
- Database password: the password to protect Gerrit's database from unauthorized access. Specify its value when you first run the post-install script. Make sure you note the value because you will be asked for it later.

a) Restart the Gerrit services.

```
/etc/init.d/collabnet restart gerrit
```

b) To verify the Git integration, run the following command.

```
/etc/init.d/collabnet status
```

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Install TeamForge 7.0 on three boxes for intensive database utilization

In this option, we install the database and datamart (reporting database) on one box, scm (cvs/subversion/git) on the second box and other services on the main application box. This may be appropriate when intensive database utilization is a factor.

Important: There are no changes in the steps for installing TeamForge 7.0 Patch 1

Note: If the remote machine in which the datamart is running 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.

Note: In a multi-box installation of TeamForge, all machines must have the same time zone - without this, ETL will not function properly.

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

1. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.
   - See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
   - See *the SuSE Linux Enterprise Server deployment 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 7 for details.

3. Set up the SuSE repository.

   a) Register the machine with Novell.

   ```
   suse_register -i -a email=<MailID> -a regcode-sles=<Registration code>
   ```

   where `<MailID>` is the email address associated with your SuSE Linux Enterprise Server license and `<Registration code>` is the Suse Enterprise Linux license code.

   b) Move the existing installer repository out of the way.

   ```
   cd /etc/zypp/repos.d/
   mv SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo_old
   ```

   c) Make sure the correct Novell repository is enabled.

   Open `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo` in a text editor and confirm that `enabled=1`.

   d) Refresh `zypper`.

   ```
   zypper ref
   ```

   Note: As an alternative, you can insert the SuSe Linux Enterprise Server DVD.

   - Ensure that the DVD is mounted in the location specified by the *baseurl* variable in `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo`.
   - Use `zypper info postgresql` to check that the DVD is properly mounted.

4. If you are installing TeamForge with internet access:

   - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
   - Copy it to `/etc/yum.repos.d/`.
   - Refresh your repository cache.

   ```
   zypper ref
   ```

   If you are installing TeamForge without internet access:

   a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

   b) Unpack the disconnected installation package.

   ```
   rpm -ivh <package-name>
   ```

   c) **Note:** If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

   Insert the SuSE installation DVD.

   This will provide some of the necessary software and utilities for installing TeamForge.

   d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

If necessary, replace `sr0` with the identifier for your server's CD drive.

Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.

5. Install the TeamForge application, reporting service.

```
zypper install teamforge-app teamforge-etl
```

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

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

At this stage, we'll make the minimal modifications required to set up a usable TeamForge 7.0 site.

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

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) Identify the servers and the services running on them.

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

c) Configure the database and the datamart.

Tip: For more information about configuring variables, see *site-options.conf* on page 350.

```
DATABASE_TYPE=postgresql
DATABASE_USERNAME=ctfuser
DATABASE_PASSWORD=ctfpwd
DATABASE_NAME=ctfdb
DATABASE_READ_ONLY_USER=ctfrouser
DATABASE_READ_ONLY_PASSWORD=ctfropwd
DATABASE_MAX_POOL_SIZE=220
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
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

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

We also recommend the following:

- For small dataset customers: use the same instance as that of `ctfdb`; do not specify REPORTS_DATABASE_PORT.

- For medium to large dataset customers: use a separate instance; specify REPORTS_DATABASE_PORT.

d) **Important:** If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

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.

f) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

g) Don't forget to save the file.

h) Include the list of TeamForge user accounts for the tokens whose password should not expire.

```
USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer
```

If you are installing git integration, add a TeamForge user for example,"gitadmin" with site-administrator rights and append the username against this parameter.

```
USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer,gitadmin
```

7. Create the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

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

```
export http_proxy=
```

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

9. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
- See *the SuSE Linux Enterprise Server deployment guide* for help.

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

10. Install the TeamForge database packages.

```
zypper install teamforge-database
```

11. Create the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -V
```

Do this on the SCM server. We'll call this my.scmandgit.box

12. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
- See *the SuSE Linux Enterprise Server deployment guide* for help.

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

13. Install the TeamForge, source control and GIT packages.

```
zypper install teamforge-scm teamforge-git
```

14. Copy the `site-options.conf` from the application box.

```
HOST_my.host.name=app etl database datamart subversion cvs indexer gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

15. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

16. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./bootstrap-data.sh
```

17. Start TeamForge.

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

Do this on my.app.box:

18. Set up the initial site data.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./bootstrap-data.sh
```

19. 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.

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

20. Integrate the CLI reports by running the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

21. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

22. Revoke the super user permissions of database and datamart users.

```
/opt/collabnet/teamforge/runtime/scripts/revoke-superuser-permission.py
```

Do this in myscmandgit.box

23. Integrate gerrit by running the `post-install.py` script.

Tip: Skip this step if you are not installing git integration.

```
/opt/collabnet/gerrit/scripts/post-install.py
```

The post installation script will try to detect the required configuration parameters. The following 3 parameters do not have default values and you will be asked to provide them:

- Teamforge Login name: the dedicated TeamForge site administrator account that does not expire and cannot be locked
- Teamforge Password: the password for the above account
- Database password: the password to protect Gerrit's database from unauthorized access. Specify its value when you first run the post-install script. Make sure you note the value because you will be asked for it later.

a) Restart the Gerrit services.

```
/etc/init.d/collabnet restart gerrit
```

b) To verify the Git integration, run the following command.

```
/etc/init.d/collabnet status
```

Do this on my.databaseanddatamart.box

24. Run the following script to set permissions for the TeamForge database read-only user specified by the *DATABASE_READ_ONLY_USER* token.

```
/opt/collabnet/teamforge/runtime/scripts/set-ctfdb-readonly-user-
permission.py
```

25. Run the following script to set permissions for the reporting database read-only user.

```
/opt/collabnet/teamforge/runtime/scripts/set-reports-readonly-user-
permission.py
```

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Install TeamForge the advanced way

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.

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 principle, a multi-box 7.0 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 TeamForge 7.0 with source control on a separate box

In this option, we install the site database, the datamart and the reporting service on the main application box and source control services on a separate box.

Important: There are no changes in the steps for installing TeamForge 7.0 Patch 1

Note: In this example, we will use the same port for the site database and the reports database. This is the default setting. It is advised to use port 5632 for the reporting database if there is heavy traffic. See *Option 1* for an example.

Note: In a multi-box installation of TeamForge, all machines must have the same time zone - without this, ETL will not function properly.

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

1. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.
   - See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
   - See *the SuSE Linux Enterprise Server deployment 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 7 for details.
3. Set up the SuSE repository.
   a) Register the machine with Novell.

   ```
   suse_register -i -a email=<MailID> -a regcode-sles=<Registration code>
   ```

   where `<MailID>` is the email address associated with your SuSE Linux Enterprise Server license and `<Registration code>` is the Suse Enterprise Linux license code.

   b) Move the existing installer repository out of the way.

   ```
   cd /etc/zypp/repos.d/
   mv SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo_old
   ```

   c) Make sure the correct Novell repository is enabled.

   Open `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo` in a text editor and confirm that `enabled=1`.

   d) Refresh `zypper`.

   ```
   zypper ref
   ```

   Note: As an alternative, you can insert the SuSe Linux Enterprise Server DVD.
   - Ensure that the DVD is mounted in the location specified by the *baseurl* variable in `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo`.
   - Use `zypper info postgresql` to check that the DVD is properly mounted.
4. If you are installing TeamForge with internet access:
   - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
   - Copy it to `/etc/yum.repos.d/`.
   - Refresh your repository cache.

   ```
   zypper ref
   ```

   If you are installing TeamForge without internet access:

   a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

   b) Unpack the disconnected installation package.

   ```
   rpm -ivh <package-name>
   ```

c) **Note:** If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

Insert the SuSE installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

If necessary, replace `sr0` with the identifier for your server's CD drive.

Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.

5. Install the TeamForge application and reporting service.

```
zypper install teamforge-app teamforge-database teamforge-etl
```

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

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
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.

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

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

At this stage, we'll make the minimal modifications required to set up a usable TeamForge 7.0 site.

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

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) Identify the boxes and the services running on them.

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

c) Configure the database and the datamart.

Tip: For more information about configuring variables, see *site-options.conf* on page 350.

```
DATABASE_TYPE=postgresql
DATABASE_USERNAME=ctfuser
DATABASE_PASSWORD=ctfpwd
DATABASE_NAME=ctfdb
DATABASE_READ_ONLY_USER=ctfrouser
DATABASE_READ_ONLY_PASSWORD=ctfropwd
DATABASE_MAX_POOL_SIZE=220
```

```
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_MAX_POOL_SIZE=30
```

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

d)

Important: If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

e) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

f)

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.

g) Save the file after you make the necessary changes in the `site-options.conf` file.

8. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

9. 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    <DATABASE_NAME>             <DATABASE_READ_ONLY_USERNAME>
<IP address of my.app.box>/32   md5
```

```
exit
```

c) Create the database user and password that is used to connect from TeamForge.

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

d) Create the database read-only user that is used to connect from TeamForge.

```
createuser -P -S --no-createdb --no-createrole
<database_readonlyusername>
```

e) Create the database.

```
createdb -E UTF8 -O <ctfdatabase_username> <ctfdatabasename>
```

f) Restart PostgreSQL.

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

10. Set up the reporting database (datamart).

Note: For small dataset customers, use the same instance as that of TeamForge database. Do not specify the *REPORTS_DATABASE_PORT*

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 database box>'
```

```
port=5632
```

b) Configure the datamart access.

```
vi /var/lib/pgsql/9.0/reports/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    <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) Create the database user and password that is used to connect from TeamForge.

```
su - postgres
createuser -p <reports_database_port> -P -S --createdb --no-createrole
<reports_database_username>
```

d) Create the database read-only user that is used to connect from TeamForge.

```
createuser -p <reports_database_port> -P -S --no-createdb --no-
createrole <reports_database_readonlyusername>
```

e) Create the database.

```
createdb -p <reports_database_port> -E UTF8 -O <reports database
username> <reports_database_name>
```

f) Restart PostgreSQL.

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

11. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./bootstrap-data.sh
```

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

```
export http_proxy=
```

13. Run the following script to set permissions for the TeamForge database read only user specified by the *DATABASE_READ_ONLY_USER* token.

```
/opt/collabnet/teamforge/runtime/scripts/set-ctfdb-readonly-user-
permission.py
```

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

14. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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.

15. Check your basic networking setup.

See *Set up networking for your TeamForge box* on page 7 for details.

16. If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
zypper ref
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c)

Note: If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

Insert the SuSE installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

If necessary, replace `sr0` with the identifier for your server's CD drive.

Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.

17. Install the TeamForge source control service.

```
zypper 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
   ```

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

d) Save the file after you make the necessary changes in the `site-options.conf` file.

19. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

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.

Note: Set the value of the Apache configuration parameter *MaxKeepAliveRequests=10000* in the `/etc/apache2/httpd.conf` file before you restart Apache.

```
cd /etc/apache2
mv httpd.conf httpd.conf_old
cp httpd.conf.cn_new httpd.conf
cd /etc/sysconfig/
mv apache2 apache2_old
cp apache2.cn_new apache2
/etc/init.d/apache2 start
```

22. Start TeamForge.

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

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.

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

24. Integrate the CLI reports by running the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

25. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

26. Run the `post-install.py` script once the TeamForge application is up and running.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

Go to *Verify your TeamForge 7.0 installation* for recommended steps to get started with your site.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Option 2: Distribute TeamForge 7.0 services using Oracle

In this option, we will use 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.

Note: In a multi-box installation of TeamForge, all machines must have the same time zone - without this, ETL will not function properly.

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

1. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
- See *the SuSE Linux Enterprise Server deployment 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 7 for details.

3. Set up the SuSE repository.
   a) Register the machine with Novell.

      ```
      suse_register -i -a email=<MailID> -a regcode-sles=<Registration code>
      ```

      where `<MailID>` is the email address associated with your SuSE Linux Enterprise Server license and `<Registration code>` is the Suse Enterprise Linux license code.
   b) Move the existing installer repository out of the way.

      ```
      cd /etc/zypp/repos.d/
      mv SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo_old
      ```
   c) Make sure the correct Novell repository is enabled.

      Open `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo` in a text editor and confirm that `enabled=1`.
   d) Refresh `zypper`.

      ```
      zypper ref
      ```

   Note: As an alternative, you can insert the SuSe Linux Enterprise Server DVD.

   - Ensure that the DVD is mounted in the location specified by the *baseurl* variable in `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo`.
   - Use `zypper info postgresql` to check that the DVD is properly mounted.

4. If you are installing TeamForge with internet access:

   - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
   - Copy it to `/etc/yum.repos.d/`.
   - Refresh your repository cache.

     ```
     zypper ref
     ```

   If you are installing TeamForge without internet access:

   a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

   b) Unpack the disconnected installation package.

      ```
      rpm -ivh <package-name>
      ```

   c) **Note:** If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

      Insert the SuSE installation DVD.

      This will provide some of the necessary software and utilities for installing TeamForge.
   d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

      If necessary, replace `sr0` with the identifier for your server's CD drive.

   Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.

5. Install the TeamForge application and reporting service.

```
zypper install teamforge-app teamforge-etl
```

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

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
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.

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

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

At this stage, we'll make the minimal modifications required to set up a usable TeamForge 7.0 site.

> **Important:** We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

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) 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
```

c) Configure the database and the datamart.

> **Tip:** For more information about configuring variables, see *site-options.conf* on page 350.

```
DATABASE_TYPE=oracle
DATABASE_USERNAME=sitedatabaseusername
DATABASE_PASSWORD=sitedatabasepwd
DATABASE_NAME=sitedatabaseinstancename
DATABASE_MAX_POOL_SIZE=220
REPORTS_DATABASE_USERNAME=reportingdatabaseusername
REPORTS_DATABASE_PASSWORD=reportingdatabasepwd
REPORTS_DATABASE_NAME=reportingdatabaseinstancename
REPORTS_DATABASE_READ_ONLY_USER=reportingreadonlyusername
REPORTS_DATABASE_READ_ONLY_PASSWORD=reportingreadonlyuserpwd
REPORTS_DATABASE_MAX_POOL_SIZE=30
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`.

d) **Important:** If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

e) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

f) **It is mandatory to include the *SCM_DEFAULT_SHARED_SECRET* token in the `site-options.conf` file of the primary TeamForge server, and provide it with a value of 16-24 characters. Remember to use the same key in the external SCM integration server also.**

g) Save the file after you make the necessary changes in the `site-options.conf` file.

8. **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:

```
zypper install <path to oracle client rpm>
```

9. Create the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

10. 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`.

11. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
- See *the SuSE Linux Enterprise Server deployment guide* for help.

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

12. Install Oracle 11G (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.*

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

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

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

See *Set up an Oracle database* on page 238 for help.

15. 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`

16. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./bootstrap-data.sh
```

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

```
export http_proxy=
```

18. Swap in the new Apache configuration file.

Note: Set the value of the Apache configuration parameter *MaxKeepAliveRequests=10000* in the `/etc/apache2/httpd.conf` file before you restart Apache.

```
cd /etc/apache2
mv httpd.conf httpd.conf_old
cp httpd.conf.cn_new httpd.conf
cd /etc/sysconfig/
mv apache2 apache2_old
cp apache2.cn_new apache2
/etc/init.d/apache2 start
```

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

19. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
- See *the SuSE Linux Enterprise Server deployment guide* for help.

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

20. Check your basic networking setup.

See *Set up networking for your TeamForge box* on page 7 for details.

21. Set up the SuSE repository.

a) Register the machine with Novell.

```
suse_register -i -a email=<MailID> -a regcode-sles=<Registration code>
```

where `<MailID>` is the email address associated with your SuSE Linux Enterprise Server license and `<Registration code>` is the Suse Enterprise Linux license code.

b) Move the existing installer repository out of the way.

```
cd /etc/zypp/repos.d/
mv SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo SUSE-Linux-
Enterprise-Server-11-SP1\ 11.1.1-1.152.repo_old
```

c) Make sure the correct Novell repository is enabled.

Open `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo` in a text editor and confirm that `enabled=1`.

d) Refresh `zypper`.

```
zypper ref
```

Note: As an alternative, you can insert the SuSe Linux Enterprise Server DVD.

- Ensure that the DVD is mounted in the location specified by the *baseurl* variable in `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo`.

- Use `zypper info postgresql` to check that the DVD is properly mounted.

22. If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
zypper ref
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) 👉 **Note:** If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

Insert the SuSE installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

If necessary, replace `sr0` with the identifier for your server's CD drive.

Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.

23. Install the TeamForge source control service.

```
zypper install teamforge-scm
```

24. 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) Don't forget to save the file.

25. Create the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

26. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./bootstrap-data.sh
```

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

```
export http_proxy=
```

28. Swap in the new Apache configuration file.

Note: Set the value of the Apache configuration parameter *MaxKeepAliveRequests=10000* in the `/etc/apache2/httpd.conf` file before you restart Apache.

```
cd /etc/apache2
mv httpd.conf httpd.conf_old
cp httpd.conf.cn_new httpd.conf
cd /etc/sysconfig/
mv apache2 apache2_old
cp apache2.cn_new apache2
/etc/init.d/apache2 start
```

29. Start TeamForge.

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

Do this on `my.app.box`

30. 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.

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

31. Integrate the CLI reports by running the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

32. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

Go to *Verify your TeamForge 7.0 installation* for recommended steps to get started with your site.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Option 3: Distribute TeamForge 7.0 services a different way

In this option, 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.

Important: There are no changes in the steps for installing TeamForge 7.0 Patch 1

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. It is advised to use port 5632 for the reporting database if there is heavy traffic.

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.

Note: In a multi-box installation of TeamForge, all machines must have the same time zone - without this, ETL will not function properly.

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

1. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.
   - See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
   - See *the SuSE Linux Enterprise Server deployment 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 7 for details.
3. Set up the SuSE repository.
   a) Register the machine with Novell.

   ```
   suse_register -i -a email=<MailID> -a regcode-sles=<Registration code>
   ```

   where `<MailID>` is the email address associated with your SuSE Linux Enterprise Server license and `<Registration code>` is the Suse Enterprise Linux license code.

   b) Move the existing installer repository out of the way.

   ```
   cd /etc/zypp/repos.d/
   mv SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo SUSE-Linux-
   Enterprise-Server-11-SP1\ 11.1.1-1.152.repo_old
   ```

   c) Make sure the correct Novell repository is enabled.

   Open `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo` in a text editor and confirm that `enabled=1`.

   d) Refresh `zypper`.

   ```
   zypper ref
   ```

   Note: As an alternative, you can insert the SuSe Linux Enterprise Server DVD.
   - Ensure that the DVD is mounted in the location specified by the *baseurl* variable in `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo`.
   - Use `zypper info postgresql` to check that the DVD is properly mounted.
4. If you are installing TeamForge with internet access:
   - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.

- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
zypper ref
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) **Note:** If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

Insert the SuSE installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

If necessary, replace `sr0` with the identifier for your server's CD drive.

Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.

5. Install the TeamForge application and reporting service.

```
zypper install teamforge-app teamforge-etl
```

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

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
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.

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

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

At this stage, we'll make the minimal modifications required to set up a usable TeamForge 7.0 site.

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

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) 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
```

c) Configure the database and the datamart.

Tip: For more information about configuring variables, see *site-options.conf* on page 350.

```
DATABASE_TYPE=postgresql
DATABASE_USERNAME=ctfuser
DATABASE_PASSWORD=ctfpwd
DATABASE_NAME=ctfdb
DATABASE_READ_ONLY_USER=ctfrouser
DATABASE_READ_ONLY_PASSWORD=ctfropwd
DATABASE_MAX_POOL_SIZE=220
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
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

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

d)

Important: If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

e) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

f) **It is mandatory to include the *SCM_DEFAULT_SHARED_SECRET* token in the `site-options.conf` file of the primary TeamForge server, and provide it with a value of 16-24 characters. Remember to use the same key in the external SCM integration server also.**

g) Save the file after you make the necessary changes in the `site-options.conf` file.

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

```
export http_proxy=
```

9. Create the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

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

10. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
- See *the SuSE Linux Enterprise Server deployment guide* for help.

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

11. Install PostgreSQL 9.0.13.

    See *the PostgreSQL docs* for support.

12. 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    <DATABASE_NAME>             <DATABASE_READ_ONLY_USERNAME>
    <IP address of my.app.box>/32   md5
    ```

    ```
    exit
    ```

    c) Create the database user and password that is used to connect from TeamForge.

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

    d) Create the database read-only user that is used to connect from TeamForge.

    ```
    createuser -P -S --no-createdb --no-createrole
    <database_readonlyusername>
    ```

    e) Create the database.

    ```
    createdb -E UTF8 -O <ctfdatabase_username> <ctfdatabasename>
    ```

    f) Restart PostgreSQL.

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

13. Set up the reporting database (datamart).

    Note: For small dataset customers, use the same instance as that of TeamForge database. Do not specify the *REPORTS_DATABASE_PORT*

    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 database box>'
    ```

    ```
    port=5632
    ```

    b) Configure the datamart access.

```
vi /var/lib/pgsql/9.0/reports/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    <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) Create the database user and password that is used to connect from TeamForge.

```
su - postgres
createuser -p <reports_database_port> -P -S --createdb --no-createrole
<reports_database_username>
```

d) Create the database read-only user that is used to connect from TeamForge.

```
createuser -p <reports_database_port> -P -S --no-createdb --no-
createrole <reports_database_readonlyusername>
```

e) Create the database.

```
createdb -p <reports_database_port> -E UTF8 -O <reports database
username> <reports_database_name>
```

f) Restart PostgreSQL.

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

Do this on `my.app.box`

14. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./bootstrap-data.sh
```

15. Swap in the new Apache configuration file.

Note: Set the value of the Apache configuration parameter *MaxKeepAliveRequests=10000* in the `/etc/apache2/httpd.conf` file before you restart Apache.

```
cd /etc/apache2
mv httpd.conf httpd.conf_old
cp httpd.conf.cn_new httpd.conf
cd /etc/sysconfig/
mv apache2 apache2_old
cp apache2.cn_new apache2
/etc/init.d/apache2 start
```

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

16. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
- See *the SuSE Linux Enterprise Server deployment 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 7 for details.

18. Set up the SuSE repository.

    a) Register the machine with Novell.

    ```
    suse_register -i -a email=<MailID> -a regcode-sles=<Registration code>
    ```

    where `<MailID>` is the email address associated with your SuSE Linux Enterprise Server license and `<Registration code>` is the Suse Enterprise Linux license code.

    b) Move the existing installer repository out of the way.

    ```
    cd /etc/zypp/repos.d/
    mv SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo_old
    ```

    c) Make sure the correct Novell repository is enabled.

    Open `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo` in a text editor and confirm that `enabled=1`.

    d) Refresh `zypper`.

    ```
    zypper ref
    ```

    Note: As an alternative, you can insert the SuSe Linux Enterprise Server DVD.

    - Ensure that the DVD is mounted in the location specified by the *baseurl* variable in `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo`.
    - Use `zypper info postgresql` to check that the DVD is properly mounted.

19. If you are installing TeamForge with internet access:

    - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
    - Copy it to `/etc/yum.repos.d/`.
    - Refresh your repository cache.

    ```
    zypper ref
    ```

    If you are installing TeamForge without internet access:

    a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

    b) Unpack the disconnected installation package.

    ```
    rpm -ivh <package-name>
    ```

    c) **Note:** If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

    Insert the SuSE installation DVD.

    This will provide some of the necessary software and utilities for installing TeamForge.

    d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

    If necessary, replace `sr0` with the identifier for your server's CD drive.

Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.

20. Install the TeamForge source control service.

```
zypper install teamforge-scm
```

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) Save the file after you make the necessary changes in the `site-options.conf` file.

22. Create the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

23. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./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.

Note: Set the value of the Apache configuration parameter *MaxKeepAliveRequests=10000* in the `/etc/apache2/httpd.conf` file before you restart Apache.

```
cd /etc/apache2
mv httpd.conf httpd.conf_old
cp httpd.conf.cn_new httpd.conf
cd /etc/sysconfig/
mv apache2 apache2_old
cp apache2.cn_new apache2
/etc/init.d/apache2 start
```

26. Start TeamForge.

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

Do this on `my.app.box`

27. Start TeamForge.

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

28. Integrate the CLI reports by running the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

29. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

Go to *Verify your TeamForge 7.0 installation* for recommended steps to get started with your site.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Verify your TeamForge 7.0 installation

Congratulations: you have just installed your TeamForge 7.0 site. 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 246 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 from Teamforge user interface* on page 234.

4. To verify that the tracker initial load (triggered through `TrackerInitialJob`) completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

   ```
   select * from etl_job where job_name ='tracker_initial_etl'
   ```

   The status column should have a value of 1.

5. 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 manually* on page 243.

   Note: This procedure is recommended, but not required.

6. Create a sample project.

   See *Create a TeamForge project*.

7. 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 TeamForge 7.0 site* on page 232 to help keep your site going.

Uninstall TeamForge 7.0

To remove TeamForge completely, use the zypper 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 zypper to remove TeamForge.

```
zypper remove TeamForge-installer
```

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

Install TeamForge on VMware Player or VMware ESXi

To get the functionality of CollabNet TeamForge with the ease of installation and maintenance that comes with VMware, run CollabNet TeamForge on VMware.

Note: For the hardware and software required to run TeamForge 7.0, see *Hardware and software requirements for CollabNet TeamForge on a virtual machine*.

Get TeamForge 7.0 for VMware Player or VMware ESXi

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

Install TeamForge on VMware Player

The VMware installation is ideal for small number of TeamForge user installations due to its detailed installation process. You have to download the CollabNet TeamForge installer for VMware Player and unzip it on the machine that will host your TeamForge site.

Note: The machine on which you are running the VMware Player must have at least 4 GB RAM and a 2 GHz processor.

Installing the VMware Player

1. Before you start TeamForge installation, install the VMware Player.
   - Download the VMware Player for Windows (64bit) from *vmware.com*.
   - Double-click the `VMware-player.exe` file and follow the instructions.
2. Start the VMware Player.
   - Click **Start** > **Programs** > **VMware** > **VMware Player**.

   Note: You don't have to update the VMware Player when you start the VMware Player for the first time.

Downloading the TeamForge 7.0 installer for VMware Player

3. Contact the *CollabNet Support* and download the TeamForge installer zip file.
4. Unzip the `TeamForge-7.0.0.0-VMware.zip` file.

Installing the TeamForge 7.0 on VMware Player

5. Click **Open a Virtual Machine**.
6. Browse and select the `TeamForge-70.ova` file from where you unzipped the `TeamForge-7.0.0.0-VMware.zip` file earlier.
7. Click **Import**.
   Once successfully imported, you can see a new virtual machine named "**TeamForge-70**" on the VM host list.
8. Click **Edit Virtual machine Settings** and set the following parameters.
   a) Set the **No. of Processors**.
   b) Set the **Memory** to 4 GB.
   c) Set the **MAC ID** of the network adapter.
   d) Click **Save**.
9. Select the **TeamForge-70** virtual machine and click **Play Virtual Machine**.

It may take a few minutes to start the **TeamForge-70** virtual machine depending on your computer's processing speed.

Proceed with configuring your VMware Player installation by following the steps in *Configure CollabNet TeamForge on VMware Player*.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Install TeamForge on VMware ESXi

The VMware ESXi 5.0 or later image server is ideal for large TeamForge user installations due to its simplified installation process. You can download the CollabNet TeamForge installer from CollabNet and unzip it on the machine to host your TeamForge site.

- Install the VMware ESXi 5.0 or later on the server where the Teamforge 7.0 would be hosted.
- Install the VMware vSphere Client 5.0 or later on a Windows (client) computer to connect to the VMware ESXi TeamForge server.

To install the TeamForge 7.0 on a VMware ESXi server:

1. Contact the *CollabNet Support* and download the TeamForge installer zip file.
2. Unzip the `TeamForge-7.0.0.1-VMware-xxx.zip` file.
3. Run the VMware vSphere Client and select the VMware ESXi host on which you want to install the TeamForge.
4. Select **File** > **Deploy OVF Template**.
   The **Deploy OVF Template** dialog box appears.
5. Click **Browse** and select the `TeamForge-70.ova` file from where you unzipped the `TeamForge-7.0.0.1-VMware-xxx.zip` file earlier.
6. Click **Next**.
   The **OVF Template Details** wizard appears.
7. Click **Next**.
8. Type a name, for example "TeamForge-7.0" and click **Next**.
9. Select the **Thin Provision** check box and click **Next**.
10. Clear the **Power On after Deployment** check box if it's selected by default.
11. Click **Finish**.
    Wait for the OVF template deployment to complete.
12. Click **Close** when the deployment is complete.
    Open the main VMware vSphere Client window and expand the VMware ESXi host to see the "TeamForge-7.0" virtual machine.
13. Select the "TeamForge-7.0" virtual machine.
14. Select the **Summary** tab from the work area.
15. Click **Edit Settings**.
16. Configure the hardware settings:
    a) Set the memory to 4096 MB.
    b) Set the CPU as 2 Cores.
    c) Set the hard disk memory as 100 GB.
    d) Click **OK**.
17. Start the TeamForge virtual machine: click **Power On** on the **Summary** tab.
    Select the **Console** tab to view the TeamForge virtual machine's console output.

Proceed with configuring your VMware ESXi installation by following the steps in *Configure CollabNet TeamForge on VMware Player*.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Configure CollabNet TeamForge on VMware Player

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

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

1. In the VMware Player, log in with the username `root` and the password `changeme`.
2. Enter and confirm a new Linux password.

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

   Type `q` to close it.

   Tip: You can use the space bar to advance a screen at a time.
5. If you accept the license terms, type `y`.
6. In the CollabNet TeamForge configuration tool, choose **Dynamic Networking (DHCP)** or **Static Networking (Static IP)**

   - Dynamic networking is useful for a one-person trial installation. It is quick and easy, but email integration with TeamForge will not work correctly.

     Note: If your IP address changes, you may also have to reconfigure source control.
   - Static networking is best if you are evaluating CollabNet TeamForge with a team, or if you already have a license and intend to use TeamForge to support your team.

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

     Tip: In this case, it's also a good idea to run TeamForge in VMware Player on a dedicated machine.

   The networking for TeamForge is restarted.
7. Specify your outgoing email (SMTP) server.

   - For a one-person evaluation, accept the default value.
   - If you have a CollabNet TeamForge license and intend to send email outside of your firewall, use the SMTP server settings provided by your network administrator.

     Note: Depending on your corporate email configuration, your system administrator may need to permit TeamForge to send mail to the corporate mail server.
8. Choose whether to run CollabNet TeamForge at startup.

   - Choose “Yes” to start CollabNet TeamForge automatically whenever you start the TeamForge VMware image.
   - Choose “No” to require a manual CollabNet TeamForge startup whenever you start the TeamForge VMware image.
9. At the prompt, click **Enter** to start your CollabNet TeamForge site.

   Note: Startup can take several minutes, depending on the speed of the host system. On some slower systems, you may get a false failure message from JBoss, like this: `jboss (app) (localhost:8080) .....failed to start in 600 seconds, giving up now. Please check the log: /opt/collabnet/teamforge/log/apps/service.log FAILED`

   This can safely be ignored.

   Tip: From now on, you can stop and restart TeamForge using these commands:

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

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

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

10. Log into your new site with the user name `admin` and the password `admin`.

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

    Note: You will have to change your administrator password when you first log in.

Configure the CLI Jobs Linked Application's URL

Important: Configure the CLI Jobs linked application's URL only if the site option token *HOST_localhost* is configured as "localhost" in the `site-options.conf` file.

11. After logging on to TeamForge, click the **Projects** tab.
12. Click the **Look** project.
13. Click **Project Admin** in the project navigation bar.
14. Click **Project Toolbar** from the **Project Admin Menu**.
15. Select the **Linked Applications** tab.
16. Select the **CLI Jobs** linked application's check box and click **Edit**.
17. Modify the URL: Replace the "`localhost`" with the hostname or IP address of the computer that hosts the TeamForge application.
18. Click **Save**.

Note: To track the ETL job failures, configure the token SOAP_ANONYMOUS_SHARED_SECRET in site-options.conf and re-create runtime. This token sends notifications when the ETL job fails. For more details, refer to: *SOAP_ANONYMOUS_SHARED_SECRET*.

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 234.

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:
   `25:supervillaininc:144.16.116.25.:302D02150080D7853DB3E5C6F67EABC65BD3AC17D4D35CB3Z002`

   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.

Uninstall TeamForge 7.0

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 7.0

When you upgrade to TeamForge 7.0, you can have Black Duck Code Sight on the same server as the main TeamForge application, or on a separate server. Choose the instructions that fit your case.

Note:

TeamForge 7.0 Patch 1 is available. *Click here to see the release notes*. To upgrade to TeamForge 7.0 Patch 1, see the help topics listed below.

- *Apply a patch on RHEL / CentOS* on page 226
- *Apply a patch on SuSE* on page 228

Important: Due to the index content change for the search functionality in TeamForge 6.1.1, a complete re-indexing of the site is required if you are upgrading from TeamForge 6.1 or earlier versions of TeamForge. This could take several hours and the index data could double in size.

Plan your upgrade to TeamForge 7.0

As the first step in upgrading to TeamForge 7.0, 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. "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 "dedicate"d or "advanced"?* on page 223

3. Branding changes?

   Every release of TeamForge can bring changes to the look and feel of the product. TeamForge 7.0 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.

4. 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 291 for recommendations from CollabNet's performance team on optimizing PostgreSQL for different conditions.

5. Upgraded JDK

   TeamForge 7.0 uses JDK 1.7.0. 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.

Upgrade to TeamForge 7.0 on Red Hat/CentOS

You can upgrade to TeamForge 7.0 from TeamForge 6.2. 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: The YUM installer will install the CollabNet TeamForge site in the default directory "/opt/collabnet/teamforge" only.

Note: Due to the index content change for the search functionality in TeamForge 6.1.1, a complete re-indexing of the site is required if you are upgrading from TeamForge 6.1 or earlier versions of TeamForge. This could take several hours and the index data could double in size.

Upgrade to TeamForge 7.0 on the same box - Black Duck Code Sight on a separate box

To upgrade to TeamForge 7.0, you install the new site and convert your existing site's data.

Important: There are no changes in the steps to upgrade to TeamForge 7.0 Patch 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. However, the priority here is to keep things as simple and quick as possible as we are working with a dedicated installation. See *Should I upgrade to TeamForge 7.0 on a new box?* on page 290 for some background information.

Tip: If there is any doubt about what kind of site you are working with, see *Is my TeamForge site "dedicate"d or "advanced"?* on page 223

Note: In a multi-box installation of TeamForge, all machines must have the same time zone - without this, ETL will not function properly.

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/9.0/backups/
teamforge_data_backup.dmp
exit
mkdir /tmp/backup_dir
cp /var/lib/pgsql/9.0/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/9.0/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)
/gitroot	GIT source code repositories

```
cp -Rpf /svnroot /sf-svnroot /cvsroot /opt/collabnet/teamforge/var /
tmp/backup_dir
```

If GIT integration is enabled, do the following:

```
mkdir /tmp/backup_dir/gerrit
cp -Rpf /gitroot /tmp/backup_dir
cp -Rpf /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
backup_dir/gerrit
```

If the old server has the TeamForge CLI server add-on installed, then backup the `cliserver.properties` file.

```
cp -Rpf /etc/cliserver.properties /tmp/backup_dir/
```

👉 **Note:** Skip this step if you are migrating from TeamForge 6.2.0.1

```
cd /tmp
tar czvf 62backup.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. Move the repositories from any previous installs out of the way.

```
mv /etc/yum.repos.d/collabnet-6.2.0.0.repo /etc/yum.repos.d/
collabnet-6.2.0.0.repo.cn_backup
```

4. Run the following commands to upgrade RHEL/CentOS to 6.4

```
yum erase neon-devel -y
yum upgrade
```

5. Set the TeamForge repository configuration.

If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
yum clean all
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) 👉 **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

Insert the Red Hat installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) Mount the Red Hat installation DVD.

```
cd /media/
mkdir cdrom
mount /dev/cdrom ./cdrom/
```

Replace `cdrom` with the identifier for your server's CD drive, if necessary.

👉 **Tip:** If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

6. Install the following application packages.

a) TeamForge: To install the TeamForge application packages run the following command:

```
yum install teamforge
```

b) GIT: To install the GIT packages run the following command:

```
yum install teamforge-git
```

7. In the `site-options.conf` file, make sure you do the following:

👉 **Important:** We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

a) Update the host name and domain name, if required.

HOST_my.host.name=app database subversion cvs etl datamart `indexer`

b) Configure the following settings if you're installing Git.

```
HOST_my.host.name=app etl database datamart subversion cvs
 indexer gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

Note: If an older version of TeamForge is integrated with Gerrit and if certain specific properties are customized directly in `<gerrit.config>`, the customized properties are not migrated as they were configured when you upgrade to TeamForge 7.0

- Copy the value of teamforge.jumbopushThresold if it is not equal to "30" and add the following property to`site-options.conf` *GERRIT_GIT_PUSH_THRESHOLD* =<your_current_value_of "jumboPushThreshold"_from_gerrit.config>
- Copy the value of teamforge.refreshTimeout if it is not equal to "3600" and add the following property to `site-options.conf` property *GERRIT_GIT_REFRESH_PERIOD* =<your_current_value_of_"refreshTimeout"_from_gerrit.config>

c) Check the *JAVA_HOME* variable to ensure that TeamForge is using the right JDK.

The JAVA_HOME variable should look like this:

```
JAVA_HOME=/usr/java/jdk1.7.0_21
```

d) If your site is running in SSL mode (SSL=on), add the following java runtime property to the *JBOSS_JAVA_OPTS* token.

```
JBOSS_JAVA_OPTS=-Xms1536m -Xmx1536m -XX:MaxPermSize=512m -server -XX:
+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/tmp -verbose:gc -XX:+
PrintGCTimeStamps -XX:+PrintGCDetails -Djsse.enableSNIExtension=false
 -Dsun.rmi.dgc.client.gcInterval=600000 -
Dsun.rmi.dgc.server.gcInterval=600000
```

e) **If the token *REQUIRE_PASSWORD_SECURITY* is enabled, then set a value for the token *PASSWORD_CONTROL_EFFECTIVE_DATE* to set the date from which the Password Control Kit (PCK) begins to disable, delete or expire user accounts. If the effective date is not set, the PCK starts disabling, deleting or expiring user accounts immediately.**

f)

Important: If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

g) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

h) Comment out the following tokens if they already enabled.

```
MAX_WWW_CLIENTS
```

```
PGSQL_MAX_CONNECTIONS
```

```
DATABASE_MAX_POOL_SIZE
```

```
SITE_DIR
```

i) Set the USERS_WITH_NO_EXPIRY_PASSWORD token as follows:

```
USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer
```

j) Make sure that the following tokens have a value if ETL is enabled.

```
SOAP_ANONYMOUS_SHARED_SECRET=
ETL_SOAP_SHARED_SECRET=
```

k) Configure the following settings for Black Duck Code Sight.

Note: In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>

To enable SSL for Black Duck Code Sight, include this token:
BDCS_SSL=on
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
BDCS_ADMIN_PASSWORD=<password>
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

8. Run the following command to remove the pagespeed cache.

```
cd /opt/collabnet/teamforge/cache
rm -rf pagespeed
```

9. Restore the `cliserver.properties` file.

```
cp -Rpf /tmp/backup_dir/cliserver.properties /opt/collabnet/teamforge/var/
etc/
```

10. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

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.

Do this on the Black Duck Code Sight server. We'll call this `my.code.box`.

12. Set the TeamForge repository configuration.

If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
yum clean all
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) 👉 **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

Insert the Red Hat installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) Mount the Red Hat installation DVD.

```
cd /media/
mkdir cdrom
mount /dev/cdrom ./cdrom/
```

Replace `cdrom` with the identifier for your server's CD drive, if necessary.

👉 **Tip:** If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

13. Set up the Black Duck Code Sight server using *these instructions*.

Do this on my.app.box:

14. Convert your site data to work with TeamForge 7.0.

👉 **Tip:** Before you kick off the data migration, use the `/etc/init.d/collabnet status` command to make sure the Jboss, Tomcat and Tomcatcs services 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 7.0 site.

15. Run the following post migration script.

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

16. Start TeamForge.

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

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

17. If the CLI reports are not configured with the earlier versions of TeamForge, run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

18. If the CLI reports are already configured with the earlier versions of TeamForge, follow these steps:

a) Backup the CLI reports.

```
sudo mkdir /root/backup
cd /root/backup
backup> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> sudo find . -name .svn | xargs sudo rm -rf
```

b) Remove the `cli/reports` folder from the branding repository.

Note: Do a SVN delete to remove this folder from the repository.

```
sudo mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup>sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> sudo svn delete reports
cli> sudo svn commit -m "To delete the old CLI reports folder"
```

c) Run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

d) Commit the back-up 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```
sudo mkdir /root/restore
cd /root/restore
restore> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
sudo mkdir custom-reports
sudo cp -avx /root/backup/pkg custom-reports
sudo svn add custom-reports
sudo svn commit -m "adding the existing customized reports"
```

e) Manually schedule the cron job from the CLI command prompt.

```
/opt/collabnet/teamforge/add-ons/teamforge_cli/bin/ctf
ctf > server add <alise_name> <http/https://hostname>
ctf/list(2)> Now, hit the Enter key.
ctf > conn <alise_name>
```

You may be prompted to enter the username (<admin_username>) and password (<admin_password>)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
```

```
ctf> \q
```

19. Integrate gerrit by running the `post-install.py` script.

    Tip: Skip this step if you are not installing git integration.

    ```
    /opt/collabnet/gerrit/scripts/post-install.py
    ```

    The post installation script will try to detect the required configuration parameters. The following 3 parameters do not have default values and you will be asked to provide them:

    - Teamforge Login name: the dedicated TeamForge site administrator account that does not expire and cannot be locked
    - Teamforge Password: the password for the above account
    - Database password: the password to protect Gerrit's database from unauthorized access. Specify its value when you first run the post-install script. Make sure you note the value because you will be asked for it later.

    a) Restart the Gerrit services.

    ```
    /etc/init.d/collabnet restart gerrit
    ```

    b) To verify the Git integration, run the following command.

    ```
    /etc/init.d/collabnet status
    ```

20. Run the following initial ETL load jobs.

    a) Change to the `runtime/scripts` directory.

    ```
    cd /opt/collabnet/teamforge/runtime/scripts
    ```

    b) Run the `TrackerInitialJob`.

    ```
    ./etl-client.py -r TrackerInitialJob
    ```

    c) Run the `SCMInitialJob`.

    ```
    ./etl-client.py -r SCMCommitInitialJob
    ```

    Tip: For more information see *When do I run the initial load job?* on page 294.

21. Integrate Black Duck Code Sight with TeamForge by running the Black Duck Code Sight `post-install.sh` script.

    Note: It is assumed that Subversion's client configuration file (`/root/.subversion/config`) for the root user is the default one without customization.

    Note: If the Black Duck Code Sight is running on a separate server, run the following command in the code sight server.

    ```
    sudo /opt/collabnet/teamforge/runtime/scripts/codesearch/post-install.sh
    ```

22. After you run the Black Duck Code Sight `post-install.sh` script, run the following script from the same directory:

    ```
    ./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
    ```

    Provide a repository base URL similar to "http://myint.box.net/svn/repos" where `myint.box` is the box with the SCM integration server.

    Also, if you add a new integration server at some point later, you'll need to run this script, (after creating new integration server), in the TeamForge `app` box.

    Important: Do this if you have a multi-box setup with a SCM integration server on a separate server.

23. If the token *VALIDATE_SSL_CERTS* is set to "true", you must run the codesearch runtime script *trust-cert.sh* in the application box.

    ```
    cd /opt/collabnet/teamforge/runtime/scripts/codesearch
    ./trust-cert.sh
    ```

24. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.0

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

See *Synchronize TeamForge source control integrations* on page 251.

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 246.

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 239 for details.

f) To verify that the tracker initial load (triggered through `TrackerInitialJob`)is completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

```
select * from etl_job where job_name ='tracker_initial_etl'
```

The status column should have a value of 1.

g) To verify that the SCM initial load (triggered through `SCMCommitInitialJob`)is completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

```
select * from etl_job where job_name ='scm_commit_initial_etl'
```

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

See *Customize anything on your site*.

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

See *Create a site-wide broadcast*.

Important: Do not delete the `teamforge-installer/7.0.0.1` 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.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Upgrade to TeamForge 7.0 on new hardware - Black Duck Code Sight on same box

To upgrade to TeamForge 7.0, set up a new box, then bring over your old site's data and convert it.

Important: There are no changes in the steps to upgrade to TeamForge 7.0 Patch 1

TeamForge 7.0 runs on Red Hat 6.3/6.4. To take advantage of the new operating system, you should install a fresh box and move your TeamForge site to it.

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

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/9.0/backups/
teamforge_data_backup.dmp
exit
mkdir /tmp/backup_dir
cp /var/lib/pgsql/9.0/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/9.0/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)
/gitroot	GIT source code repositories

```
cp -Rpf /svnroot /sf-svnroot /cvsroot /opt/collabnet/teamforge/var /
tmp/backup_dir
```

If GIT integration is enabled, do the following:

```
mkdir /tmp/backup_dir/gerrit
cp -Rpf /gitroot /tmp/backup_dir
cp -Rpf /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
backup_dir/gerrit
```

If the old server has the TeamForge CLI server add-on installed, then backup the `cliserver.properties` file.

Note: Skip this step if you are migrating from TeamForge 6.2.0.1

```
cp -Rpf /etc/cliserver.properties /tmp/backup_dir/
```

```
cd /tmp
tar czvf 62backup.tgz backup_dir
```

c) Back up your SSH keys, if any.

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

2. Stop the TeamForge application server and the Apache server.

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

3. Stop PostgreSQL.

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

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

```
scp /opt/collabnet/teamforge-installer/6.2.0.0/conf/site-options.conf
username@newbox:/tmp
```

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

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

```
scp /tmp/62backup.tgz username@newbox:/tmp
```

Do this on the new TeamForge 7.0 box:

6. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.
   - The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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. Check your basic networking setup.

   See *Set up networking for your TeamForge box* on page 7 for details.

8. Set the TeamForge repository configuration.

   If you are installing TeamForge with internet access:

   - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
   - Copy it to `/etc/yum.repos.d/`.
   - Refresh your repository cache.

     ```
     yum clean all
     ```

   If you are installing TeamForge without internet access:

   a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

   b) Unpack the disconnected installation package.

   ```
   rpm -ivh <package-name>
   ```

   c) **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

   Insert the Red Hat installation DVD.

   This will provide some of the necessary software and utilities for installing TeamForge.

   d) Mount the Red Hat installation DVD.

   ```
   cd /media/
   mkdir cdrom
   mount /dev/cdrom ./cdrom/
   ```

   Replace `cdrom` with the identifier for your server's CD drive, if necessary.

   Tip: If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

   e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

   f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
```

```
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

9. Install the following application packages.

a) TeamForge: To install the TeamForge application packages run the following command:

```
yum install teamforge
```

b) Black Duck Code Sight: To install Black Duck Code Sight run the following command:

```
yum install teamforge-codesearch
```

c) GIT: To install the GIT packages run the following command:

```
yum install teamforge-git
```

10. Copy the `site-options.conf` file to the TeamForge installer directory.

```
cp /tmp/site-options.conf /opt/collabnet/teamforge-installer/7.0.0.1/conf
```

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

```
cd /tmp
tar xzvf 62backup.tgz
```

12. In the `site-options.conf` file, make sure you do the following:

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

a) Update the host name and domain name, if required.

HOST_my.host.name=app database subversion cvs etl datamart `indexer`

b) If you're installing GIT, configure the following settings:

```
HOST_my.host.name=app etl database datamart subversion cvs
 indexer gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

Note: If an older version of TeamForge is integrated with Gerrit and if certain specific properties are customized directly in `gerrit.config>`, the customized properties are not migrated as they were configured when you upgrade to TeamForge 7.0

- Copy the value of teamforge.jumbopushThresold if it is not equal to "30" and add the following property to`site-options.conf` *GERRIT_GIT_PUSH_THRESHOLD* =<your_current_value_of "jumboPushThreshold"_from_gerrit.config>
- Copy the value of teamforge.refreshTimeout if it is not equal to "3600" and add the following property to `site-options.conf` property *GERRIT_GIT_REFRESH_PERIOD* =<your_current_value_of_"refreshTimeout"_from_gerrit.config>

c) Check the*JAVA_HOME* variable to ensure that TeamForge is using the right JDK.

The JAVA_HOME variable should look like this:

```
JAVA_HOME=/usr/java/jdk1.7.0_21
```

d) If your site is running in SSL mode (SSL=on), add the following java runtime property to the *JBOSS_JAVA_OPTS* token.

```
JBOSS_JAVA_OPTS=-Xms1536m -Xmx1536m -XX:MaxPermSize=512m -server -XX:
+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/tmp -verbose:gc -XX:+
PrintGCTimeStamps -XX:+PrintGCDetails -Djsse.enableSNIExtension=false
 -Dsun.rmi.dgc.client.gcInterval=600000 -
Dsun.rmi.dgc.server.gcInterval=600000
```

e) **If the token *REQUIRE_PASSWORD_SECURITY* is enabled, then set a value for the token *PASSWORD_CONTROL_EFFECTIVE_DATE* to set the date from which the Password Control Kit (PCK) begins to disable, delete or expire user accounts. If the effective date is not set, the PCK starts disabling, deleting or expiring user accounts immediately.**

f) 👉 **Important:** If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

g) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

h) Add the token *DEDICATED_INSTALL=true* if it is not present in the `site-options.conf` file.

The dedicated installation mode is suitable when the server is completely dedicated to TeamForge application. This makes the installation process very simple as theTeamForge installer takes care of configuring your apache, postgresql automatically without any manual intervention. In a "Dedicated" install, it is advised not to install another software application other than TeamForge. It is also recommended not to run any unwanted services that are not used by TeamForge.

i) Comment out the following tokens if they already enabled.

```
MAX_WWW_CLIENTS
```

```
PGSQL_MAX_CONNECTIONS
```

```
DATABASE_MAX_POOL_SIZE
```

```
SITE_DIR
```

j) Set the USERS_WITH_NO_EXPIRY_PASSWORD token as follows:

```
USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer
```

k) Make sure that the following tokens have a value if ETL is enabled.

```
SOAP_ANONYMOUS_SHARED_SECRET=
ETL_SOAP_SHARED_SECRET=
```

l) Configure the following settings for Black Duck Code Sight.

```
HOST_mycodesearchbox.domain.com= app database subversion cvs etl
 datamart indexer gerrit codesearch
```

👉 **Note:** In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>

To enable SSL for Black Duck Code Sight, include this token:
```

```
BDCS_SSL=on
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
BDCS_ADMIN_PASSWORD=<password>
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

13. Restore the `cliserver.properties` file.

```
cp -Rpf /tmp/backup_dir/cliserver.properties /opt/collabnet/teamforge/var/
etc/
```

14. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

15. 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`, `gitroot` and `var`.

```
cp -Rpf /tmp/backup_dir/svnroot /svnroot
cp -Rpf /tmp/backup_dir/cvsroot /cvsroot
cp -Rpf /tmp/backup_dir/sf-svnroot /sf-svnroot
cp -Rpf /tmp/backup_dir/var /opt/collabnet/teamforge/var
```

If GIT integration is enabled:

```
cp -Rpf /tmp/backup_dir/gitroot /gitroot
cp -Rpf /tmp/backup_dir/gerrit/etc /opt/collabnet/gerrit/
cp -Rpf /tmp/backup_dir/gerrit/.ssh /opt/collabnet/gerrit/
```

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 7.0.

Tip: Before you kick off the data migration, use the `/etc/init.d/collabnet status` command to make sure the Jboss, Tomcat and Tomcatcs services 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 7.0 site.

18. Run the following post migration script.

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

19. Start TeamForge.

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

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

20. If the CLI reports are not configured with the earlier versions of TeamForge, run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

21. If the CLI reports are already configured with the earlier versions of TeamForge, follow these steps:

a) Backup the CLI reports.

```
sudo mkdir /root/backup
cd /root/backup
backup> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> sudo find . -name .svn | xargs sudo rm -rf
```

b) Remove the `cli/reports` folder from the branding repository.

Note: Do a SVN delete to remove this folder from the repository.

```
sudo mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup>sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> sudo svn delete reports
cli> sudo svn commit -m "To delete the old CLI reports folder"
```

c) Run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

d) Commit the back-up 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```
sudo mkdir /root/restore
cd /root/restore
restore> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
sudo mkdir custom-reports
sudo cp -avx /root/backup/pkg custom-reports
sudo svn add custom-reports
sudo svn commit -m "adding the existing customized reports"
```

e) Manually schedule the cron job from the CLI command prompt.

```
/opt/collabnet/teamforge/add-ons/teamforge_cli/bin/ctf
ctf > server add <alise_name> <http/https://hostname>
ctf/list(2)> Now, hit the Enter key.
ctf > conn <alise_name>
```

You may be prompted to enter the username (<admin_username>) and password (<admin_password>)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```

22. Integrate Gerrit by running the `post-install.py` script.

```
/opt/collabnet/gerrit/scripts/post-install.py
```

a) Restart the Gerrit services.

```
/etc/init.d/collabnet restart gerrit
```

b) To verify the GIT integration:

Login to the app server and run the following command:

```
/etc/init.d/collabnet status
```

23. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

24. Integrate Black Duck Code Sight with TeamForge by running the Black Duck Code Sight `post-install.sh` script.

Note: It is assumed that Subversion's client configuration file (`/root/.subversion/config`) for the root user is the default one without customization.

Note: If the Black Duck Code Sight is running on a separate server, run the following command in the code sight server.

```
sudo /opt/collabnet/teamforge/runtime/scripts/codesearch/post-install.sh
```

25. After you run the Black Duck Code Sight `post-install.sh` script, run the following script from the same directory:

```
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL similar to "http://myint.box.net/svn/repos" where `myint.box` is the box with the SCM integration server.

Also, if you add a new integration server at some point later, you'll need to run this script, (after creating new integration server), in the TeamForge `app` box.

Important: Do this if you have a multi-box setup with a SCM integration server on a separate server.

26. If the token *VALIDATE_SSL_CERTS* is set to "true", you must run the codesearch runtime script *trust-cert.sh* in the application box.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
```

27. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.0

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

See *Synchronize TeamForge source control integrations* on page 251.

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 246.

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 239 for details.

f) To verify that the tracker initial load (triggered through `TrackerInitialJob`)is completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

```
select * from etl_job where job_name ='tracker_initial_etl'
```

   The status column should have a value of 1.

g) To verify that the SCM initial load (triggered through `SCMCommitInitialJob`)is completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

```
select * from etl_job where job_name ='scm_commit_initial_etl'
```

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

   See *Customize anything on your site*.

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

   See *Create a site-wide broadcast*.

Important: Do not delete the `teamforge-installer/7.0.0.1` 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.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Upgrade to TeamForge 7.0 on the same hardware - SCM on a separate box

To upgrade to TeamForge 7.0, you install the new site and convert your existing site's data.

Important: There are no changes in the steps to upgrade to TeamForge 7.0 Patch 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. However, the priority here is to keep things as simple and quick as possible as we are working with a dedicated installation. See *Should I upgrade to TeamForge 7.0 on a new box?* on page 290 for some background information.

Tip: If there is any doubt about what kind of site you are working with, see *Is my TeamForge site "dedicate"d or "advanced"?* on page 223

Note: In a multi-box installation of TeamForge, all machines must have the same time zone - without this, ETL will not function properly.

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/9.0/backups/
teamforge_data_backup.dmp
exit
mkdir /tmp/backup_dir
cp /var/lib/pgsql/9.0/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/9.0/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
/sf-svnroot	Subversion repository for branding data

```
cp -Rpf /sf-svnroot /opt/collabnet/teamforge/var /tmp/backup_dir
```

If the old server has the TeamForge CLI server add-on installed, then backup the `cliserver.properties` file.

Note: Skip this step if you are migrating from TeamForge 6.2.0.1

```
cp -Rpf /etc/cliserver.properties /tmp/backup_dir/
cd /tmp
tar czvf 62backup.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. Move the repositories from any previous installs out of the way.

```
mv /etc/yum.repos.d/collabnet-6.2.0.0.repo /etc/yum.repos.d/
collabnet-6.2.0.0.repo.cn_backup
```

4. Run the following commands to upgrade RHEL/CentOS to 6.4

```
yum erase neon-devel -y
yum upgrade
```

5. Set the TeamForge repository configuration.

If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
yum clean all
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

Insert the Red Hat installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) Mount the Red Hat installation DVD.

```
cd /media/
mkdir cdrom
mount /dev/cdrom ./cdrom/
```

Replace `cdrom` with the identifier for your server's CD drive, if necessary.

Tip: If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

6. Install the following application packages.

a) TeamForge: To install the TeamForge application packages run the following command:

```
yum install teamforge
```

7. In the `site-options.conf` file, make sure you do the following:

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

a) Update the host name and domain name, if required.

```
HOST_my.host.name=app etl database datamart indexer
```

```
HOST_myscmbox.domain.name=subversion cvs
```

b) Check the*JAVA_HOME* variable to ensure that TeamForge is using the right JDK.

The JAVA_HOME variable should look like this:

```
JAVA_HOME=/usr/java/jdk1.7.0_21
```

c) If your site is running in SSL mode (SSL=on), add the following java runtime property to the *JBOSS_JAVA_OPTS* token.

```
JBOSS_JAVA_OPTS=-Xms1536m -Xmx1536m -XX:MaxPermSize=512m -server -XX:
+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/tmp -verbose:gc -XX:+
PrintGCTimeStamps -XX:+PrintGCDetails -Djsse.enableSNIExtension=false
 -Dsun.rmi.dgc.client.gcInterval=600000 -
Dsun.rmi.dgc.server.gcInterval=600000
```

d) **If the token *REQUIRE_PASSWORD_SECURITY* is enabled, then set a value for the token *PASSWORD_CONTROL_EFFECTIVE_DATE* to set the date from which the Password Control Kit (PCK) begins to disable, delete or expire user accounts. If the effective date is not set, the PCK starts disabling, deleting or expiring user accounts immediately.**

e)

Important: If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

f) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

g) Add the token *DEDICATED_INSTALL=true*

This makes the installation process very simple as theTeamForge installer takes care of configuring the apache, postgresql automatically without any manual intervention.

h) Comment out the following tokens if they already enabled.

```
MAX_WWW_CLIENTS
```

```
PGSQL_MAX_CONNECTIONS
```

```
DATABASE_MAX_POOL_SIZE
```

```
SITE_DIR
```

i) Set the USERS_WITH_NO_EXPIRY_PASSWORD token as follows:

```
USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer
```

j) Make sure that the following tokens have a value if ETL is enabled.

```
SOAP_ANONYMOUS_SHARED_SECRET=
ETL_SOAP_SHARED_SECRET=
```

k) Configure the following settings for Black Duck Code Sight.

```
HOST_my.host.name=app etl database datamart indexer codesearch
```

Note: In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>

To enable SSL for Black Duck Code Sight, include this token:
```

```
BDCS_SSL=on
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
BDCS_ADMIN_PASSWORD=<password>
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

8. Run the following command to remove the pagespeed cache.

```
cd /opt/collabnet/teamforge/cache
rm -rf pagespeed
```

9. Restore the `cliserver.properties` file.

```
cp -Rpf /tmp/backup_dir/cliserver.properties /opt/collabnet/teamforge/var/
etc/
```

10. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

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 7.0.

Tip: Before you kick off the data migration, use the `/etc/init.d/collabnet status` command to make sure the Jboss, Tomcat and Tomcatcs services 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 7.0 site.

13. Run the following post migration script.

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

Do the following to upgrade the SCM box (CVS/Subversion/Git)

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.

14. Back up your SCM 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).
/gitroot	GIT source code repositories.

```
mkdir -p /tmp/scmbackup_dir/gerrit
```

```
cp -Rpf /svnroot /cvsroot /tmp/scmbackup_dir
```

If GIT integration is enabled:

```
mkdir /tmp/scmbackup_dir/gerrit
cp -Rpf /gitroot /tmp/scmbackup_dir
cp -Rpf /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
scmbackup_dir/gerrit
cd/tmp
tar czvf 62scmbackup.tgz scmbackup_dir
```

b) Back up your SSH keys, if any.

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

15. Stop TeamForge

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

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

```
mv /etc/yum.repos.d/collabnet-6.2.0.0.repo /etc/yum.repos.d/
collabnet-6.2.0.0.repo.cn_backup
```

17. Set the TeamForge repository configuration.

If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
yum clean all
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) 👉 **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

Insert the Red Hat installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) Mount the Red Hat installation DVD.

```
cd /media/
mkdir cdrom
mount /dev/cdrom ./cdrom/
```

Replace `cdrom` with the identifier for your server's CD drive, if necessary.

👉 **Tip:** If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

18. On the source code box, install the source code component of the TeamForge application.

```
yum clean all
yum install teamforge-scm
```

a) GIT: To install the GIT packages run the following command:

```
yum install teamforge-git
```

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

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

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

a) Add the token *DEDICATED_INSTALL=true*

This makes the installation process very simple as theTeamForge installer takes care of configuring the apache, postgresql automatically without any manual intervention.

b) Use the *SCM_DEFAULT_SHARED_SECRET* on page 382 key from the `site-options.conf` file of the primary TeamForge server

c) Comment out the following token if it is already enabled.

```
MAX_WWW_CLIENTS
```

d) If you're installing GIT, configure the following settings:

```
HOST_myappbox.domain.name=app etl database datamart indexer
```

```
HOST_myscmbox.host.name=subversion cvs gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

Note: If an older version of TeamForge is integrated with Gerrit and if certain specific properties are customized directly in `gerrit.config>`, the customized properties are not migrated as they were configured when you upgrade to TeamForge 7.0

- Copy the value of teamforge.jumbopushThresold if it is not equal to "30" and add the following property to`site-options.conf` *GERRIT_GIT_PUSH_THRESHOLD* =<your_current_value_of "jumboPushThreshold"_from_gerrit.config>
- Copy the value of teamforge.refreshTimeout if it is not equal to "3600" and add the following property to `site-options.conf` property *GERRIT_GIT_REFRESH_PERIOD* =<your_current_value_of_"refreshTimeout"_from_gerrit.config>

20. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

21. 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.

22. Start TeamForge

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

23. Start TeamForge.

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

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

24. Integrate Gerrit by running the `post-install.py` script.

```
/opt/collabnet/gerrit/scripts/post-install.py
```

a) Restart the Gerrit services.

```
/etc/init.d/collabnet restart gerrit
```

b) To verify the GIT integration:

Login to the app server and run the following command:

```
/etc/init.d/collabnet status
```

25. If the CLI reports are not configured with the earlier versions of TeamForge, run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

26. If the CLI reports are already configured with the earlier versions of TeamForge, follow these steps:

a) Backup the CLI reports.

```
sudo mkdir /root/backup
cd /root/backup
backup> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> sudo find . -name .svn | xargs sudo rm -rf
```

b) Remove the `cli/reports` folder from the branding repository.

Note: Do a SVN delete to remove this folder from the repository.

```
sudo mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup>sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> sudo svn delete reports
cli> sudo svn commit -m "To delete the old CLI reports folder"
```

c) Run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

d) Commit the back-up 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```
sudo mkdir /root/restore
cd /root/restore
restore> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
sudo mkdir custom-reports
sudo cp -avx /root/backup/pkg custom-reports
sudo svn add custom-reports
sudo svn commit -m "adding the existing customized reports"
```

e) Manually schedule the cron job from the CLI command prompt.

```
/opt/collabnet/teamforge/add-ons/teamforge_cli/bin/ctf
ctf > server add <alise_name> <http/https://hostname>
ctf/list(2)> Now, hit the Enter key.
ctf > conn <alise_name>
```

You may be prompted to enter the username (<admin_username>) and password (<admin_password>)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```

27. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

28. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.0

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

See *Synchronize TeamForge source control integrations* on page 251.

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 246.

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 239 for details.

f) To verify that the tracker initial load (triggered through `TrackerInitialJob`)is completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

```
select * from etl_job where job_name ='tracker_initial_etl'
```

The status column should have a value of 1.

g) To verify that the SCM initial load (triggered through `SCMCommitInitialJob`)is completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

```
select * from etl_job where job_name ='scm_commit_initial_etl'
```

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

See *Customize anything on your site*.

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

See *Create a site-wide broadcast*.

Important: Do not delete the `teamforge-installer/7.0.0.1` directory. You will need it for future maintenance and upgrades.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Upgrade to TeamForge 7.0 on new hardware - SCM on a separate box

To upgrade to TeamForge 7.0, you install the new site and convert your existing site's data.

Important: There are no changes in the steps to upgrade to TeamForge 7.0 Patch 1

TeamForge 7.0 runs on Red Hat Enterprise Linux 6.3 and 6.4. To take advantage of the new operating system, you should install a fresh box and move your TeamForge 6.2 site to it.

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/9.0/backups/
teamforge_data_backup.dmp
exit
mkdir /tmp/backup_dir
cp /var/lib/pgsql/9.0/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/9.0/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
/sf-svnroot	Subversion repository for branding data.

If the old server has the TeamForge CLI server add-on installed, then backup the `cliserver.properties` file.

Note: Skip this step if you are migrating from TeamForge 6.2.0.1

```
cp -Rpf /sf-svnroot /opt/collabnet/teamforge/var /tmp/backup_dir
cd /tmp
tar czvf 62backup.tgz backup_dir
```

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

2. Stop TeamForge application server and Apache server.

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

3. Stop PostgreSQL.

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

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

```
scp /opt/collabnet/teamforge-installer/6.2.0.0/conf/site-options.conf
username@newbox:/tmp
```

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

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

```
scp /tmp/62backup.tgz username@newbox:/tmp
```

Do this on the old TeamForge 6.2 source control box:

6. 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).
/gitroot	GIT source code repositories.

```
mkdir -p /tmp/scmbackup_dir/gerrit
cp -Rpf /svnroot /cvsroot /tmp/scmbackup_dir
```

If GIT integration is enabled:

```
cp -Rpf /gitroot /tmp/scmbackup_dir
cp -Rpf /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
scmbackup_dir/gerrit
cd /tmp
tar czvf 62scmbackup.tgz scmbackup_dir
```

7. Stop TeamForge

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

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

```
scp /opt/collabnet/teamforge-installer/6.2.0.0/conf/site-options.conf
username@newbox:/tmp
```

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

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

```
scp /tmp/62scmbackup.tgz username@newscmbox:/tmp
```

Do this on the new TeamForge 7.0 app box

10. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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. Check your basic networking setup.

    See *Set up networking for your TeamForge box* on page 7 for details.

12. Set the TeamForge repository configuration.

    If you are installing TeamForge with internet access:

    - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
    - Copy it to `/etc/yum.repos.d/`.
    - Refresh your repository cache.

      ```
      yum clean all
      ```

    If you are installing TeamForge without internet access:

    a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

    b) Unpack the disconnected installation package.

    ```
    rpm -ivh <package-name>
    ```

    c) **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

    Insert the Red Hat installation DVD.

    This will provide some of the necessary software and utilities for installing TeamForge.

    d) Mount the Red Hat installation DVD.

    ```
    cd /media/
    mkdir cdrom
    mount /dev/cdrom ./cdrom/
    ```

    Replace `cdrom` with the identifier for your server's CD drive, if necessary.

    Tip: If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

    e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

    f) Copy this into the yum configuration file:

    ```
    [RHEL-CDROM]
    name=RHEL CDRom
    baseurl=file:///media/cdrom/Server/
    gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
    enabled=1
    gpgcheck=0
    ```

    g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

13. Install the following application packages.
    a) TeamForge: To install the TeamForge application packages run the following command:

    ```
    yum install teamforge
    ```

    b) Black Duck Code Sight: To install Black Duck Code Sight run the following command:

    ```
    yum install teamforge-codesearch
    ```

14. Copy the `site-options.conf` file to the TeamForge installer directory.

    ```
    cp /tmp/site-options.conf /opt/collabnet/teamforge-installer/7.0.0.1/conf
    ```

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

    ```
    tar xzvf 62backup.tgz
    cd /tmp
    cp -Rpf /tmp/backup_dir/sf-svnroot /sf-svnroot
    cp -Rpf /tmp/backup_dir/var /opt/collabnet/teamforge/var
    ```

16. Restore the `cliserver.properties` file.

    ```
    cp -Rpf /tmp/backup_dir/cliserver.properties /opt/collabnet/teamforge/var/
    etc/
    ```

17. In the `site-options.conf` file, make sure you do the following:

    Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

    a) Update the host name and domain name, if required.

    HOST_my.host.name=app database subversion cvs etl datamart `indexer`

    b) Check the*JAVA_HOME* variable to ensure that TeamForge is using the right JDK.

    The JAVA_HOME variable should look like this:

    ```
    JAVA_HOME=/usr/java/jdk1.7.0_21
    ```

    c) If your site is running in SSL mode (SSL=on), add the following java runtime property to the *JBOSS_JAVA_OPTS* token.

    ```
    JBOSS_JAVA_OPTS=-Xms1536m -Xmx1536m -XX:MaxPermSize=512m -server -XX:
    +HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/tmp -verbose:gc -XX:+
    PrintGCTimeStamps -XX:+PrintGCDetails -Djsse.enableSNIExtension=false
     -Dsun.rmi.dgc.client.gcInterval=600000 -
    Dsun.rmi.dgc.server.gcInterval=600000
    ```

    d) **If the token *REQUIRE_PASSWORD_SECURITY* is enabled, then set a value for the token *PASSWORD_CONTROL_EFFECTIVE_DATE* to set the date from which the Password Control Kit (PCK) begins to disable, delete or expire user accounts. If the effective date is not set, the PCK starts disabling, deleting or expiring user accounts immediately.**

    e)

    Important: If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

    Comment out the token if site-wide reports is not enabled.

    f) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

g) Add the token *DEDICATED_INSTALL=true* if it is not present in the `site-options.conf` file.

The dedicated installation mode is suitable when the server is completely dedicated to TeamForge application. This makes the installation process very simple as theTeamForge installer takes care of configuring your apache, postgresql automatically without any manual intervention. In a "Dedicated" install, it is advised not to install another software application other than TeamForge. It is also recommended not to run any unwanted services that are not used by TeamForge.

h) Comment out the following tokens if they already enabled.

```
MAX_WWW_CLIENTS
```

```
PGSQL_MAX_CONNECTIONS
```

```
DATABASE_MAX_POOL_SIZE
```

```
SITE_DIR
```

i) Set the USERS_WITH_NO_EXPIRY_PASSWORD token as follows:

```
USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer
```

j) Make sure that the following tokens have a value if ETL is enabled.

```
SOAP_ANONYMOUS_SHARED_SECRET=
ETL_SOAP_SHARED_SECRET=
```

k) Configure the following settings for Black Duck Code Sight.

```
HOST_mycodesearchbox.domain.com=codesearch
```

Note: In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>

To enable SSL for Black Duck Code Sight, include this token:
BDCS_SSL=on
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
BDCS_ADMIN_PASSWORD=<password>
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

18. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

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 7.0.

Tip: Before you kick off the data migration, use the `/etc/init.d/collabnet status` command to make sure the Jboss, Tomcat and Tomcatcs services 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 7.0 site.

21. Run the following post migration script.

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

Do this on the new TeamForge 7.0 SCM box

22. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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.

23. Check your basic networking setup.

See *Set up networking for your TeamForge box* on page 7 for details.

24. Set the TeamForge repository configuration.

If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
yum clean all
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

Insert the Red Hat installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) Mount the Red Hat installation DVD.

```
cd /media/
mkdir cdrom
mount /dev/cdrom ./cdrom/
```

Replace `cdrom` with the identifier for your server's CD drive, if necessary.

Tip: If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

25. On the source code box, install the source code component of the TeamForge application.

```
yum clean all
yum install teamforge-scm
```

a) GIT: To install the GIT packages run the following command:

```
yum install teamforge-git
```

26. Reload svnroot, cvsroot and gitroot.

```
tar zxvf /tmp/62scmbackup.tgz
cp -Rpf /tmp/scmbackup_dir/svnroot /svnroot
cp -Rpf /tmp/scmbackup_dir/cvsroot /cvsroot
```

If GIT integration is enabled, do the following:

```
cp -Rpf /tmp/scmbackup_dir/gitroot /gitroot
cp -Rpf /tmp/scmbackup_dir/gerrit/etc /opt/collabnet/gerrit/
cp -Rpf /tmp/scmbackup_dir/gerrit/.ssh /opt/collabnet/gerrit/
```

27. Copy the `site-options.conf` file to the TeamForge installer directory.

```
cp /tmp/site-options.conf /opt/collabnet/teamforge-installer/7.0.0.1/conf
```

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

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

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

a) Add the token *DEDICATED_INSTALL=true*

This makes the installation process very simple as theTeamForge installer takes care of configuring the apache, postgresql automatically without any manual intervention.

b) Use the *SCM_DEFAULT_SHARED_SECRET* on page 382 key from the `site-options.conf` file of the primary TeamForge server

c) Comment out the following token if it is already enabled.

```
MAX_WWW_CLIENTS
```

d) If you're installing GIT, configure the following settings:

```
HOST_myappbox.domain.name=app etl database datamart indexer
```

```
HOST_myscmbox.host.name=subversion cvs gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

Note: If an older version of TeamForge is integrated with Gerrit and if certain specific properties are customized directly in `gerrit.config>`, the customized properties are not migrated as they were configured when you upgrade to TeamForge 7.0

- Copy the value of teamforge.jumbopushThresold if it is not equal to "30" and add the following property to`site-options.conf` *GERRIT_GIT_PUSH_THRESHOLD* =<your_current_value_of "jumboPushThreshold"_from_gerrit.config>
- Copy the value of teamforge.refreshTimeout if it is not equal to "3600" and add the following property to `site-options.conf` property *GERRIT_GIT_REFRESH_PERIOD* =<your_current_value_of_"refreshTimeout"_from_gerrit.config>

29. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

30. 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.

31. Start TeamForge

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

Do this on my.appbox.com

32. Start TeamForge.

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

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

33. Integrate Gerrit by running the `post-install.py` script.

```
/opt/collabnet/gerrit/scripts/post-install.py
```

a) Restart the Gerrit services.

```
/etc/init.d/collabnet restart gerrit
```

b) To verify the GIT integration:

Login to the app server and run the following command:

```
/etc/init.d/collabnet status
```

34. If the CLI reports are not configured with the earlier versions of TeamForge, run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

35. If the CLI reports are already configured with the earlier versions of TeamForge, follow these steps:

a) Backup the CLI reports.

```
sudo mkdir /root/backup
cd /root/backup
backup> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> sudo find . -name .svn | xargs sudo rm -rf
```

b) Remove the `cli/reports` folder from the branding repository.

Note: Do a SVN delete to remove this folder from the repository.

```
sudo mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup>sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> sudo svn delete reports
cli> sudo svn commit -m "To delete the old CLI reports folder"
```

c) Run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

d) Commit the back-up 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```
sudo mkdir /root/restore
cd /root/restore
restore> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
sudo mkdir custom-reports
sudo cp -avx /root/backup/pkg custom-reports
sudo svn add custom-reports
sudo svn commit -m "adding the existing customized reports"
```

e) Manually schedule the cron job from the CLI command prompt.

```
/opt/collabnet/teamforge/add-ons/teamforge_cli/bin/ctf
ctf > server add <alise_name> <http/https://hostname>
ctf/list(2)> Now, hit the Enter key.
ctf > conn <alise_name>
```

You may be prompted to enter the username (<admin_username>) and password (<admin_password>)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```

36. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

37. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.0

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

   See *Synchronize TeamForge source control integrations* on page 251.

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 246.

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 239 for details.

f) To verify that the tracker initial load (triggered through `TrackerInitialJob`)is completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

```
select * from etl_job where job_name ='tracker_initial_etl'
```

   The status column should have a value of 1.

g) To verify that the SCM initial load (triggered through `SCMCommitInitialJob`)is completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

```
select * from etl_job where job_name ='scm_commit_initial_etl'
```

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

   See *Customize anything on your site*.

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

   See *Create a site-wide broadcast*.

Important: Do not delete the `teamforge-installer/7.0.0.1` directory. You will need it for future maintenance and upgrades.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Upgrade an advanced TeamForge site to TeamForge 7.0

Upgrading to TeamForge 7.0 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 "dedicate"d or "advanced"?* on page 223

Upgrade TeamForge to version 7.0 on the same box (Oracle)

To upgrade to TeamForge 7.0 without changing the box where your pre TeamForge 7.0 site is running, you have to install the new TeamForge site and migrate the site data.

Important: There are no changes in the steps to upgrade to TeamForge 7.0 Patch 1

In this option, the Oracle database runs on one box, source control services runs on the second box and other services run on the application box.

Note: In a multi-box installation of TeamForge, all machines must have the same time zone - without this, ETL will not function properly.

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

1. Back up your site data.

   a) Make a dump file of your site database.

      To back up the Oracle database, follow the *Oracle backup procedure*.

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

Directory	Contents
/opt/collabnet/teamforge/var	User-created data, such as artifact attachments
/sf-svnroot	Subversion repository for branding data

```
mkdir /tmp/backup_dir
cp -Rpf /sf-svnroot /opt/collabnet/teamforge/var /tmp/backup_dir
cd /tmp
tar czvf 62backup.tgz backup_dir
```

c) Back up your SSH keys, if any.

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

2. Stop the TeamForge application server and the Apache server.

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

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

```
mv /etc/yum.repos.d/collabnet-6.2.0.0.repo /etc/yum.repos.d/
collabnet-6.2.0.0.repo.cn_backup
```

4. Run the following commands to upgrade RHEL/CentOS to 6.4

```
yum erase neon-devel -y
yum upgrade
```

5. Set the TeamForge repository configuration.

If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
yum clean all
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

Insert the Red Hat installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) Mount the Red Hat installation DVD.

```
cd /media/
mkdir cdrom
mount /dev/cdrom ./cdrom/
```

Replace `cdrom` with the identifier for your server's CD drive, if necessary.

Tip: If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

6. Install the TeamForge application.

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

7. In the `site-options.conf` file, make sure you do the following:

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

a) Update the host name and domain name, if necessary.

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

b) Check the*JAVA_HOME* variable to ensure that TeamForge is using the right JDK.

The JAVA_HOME variable should look like this:

```
JAVA_HOME=/usr/java/jdk1.7.0_21
```

c) If your site is running in SSL mode (SSL=on), add the following java runtime property to the *JBOSS_JAVA_OPTS* token.

```
JBOSS_JAVA_OPTS=-Xms1536m -Xmx1536m -XX:MaxPermSize=512m -server -XX:
+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/tmp -verbose:gc -XX:+
PrintGCTimeStamps -XX:+PrintGCDetails -Djsse.enableSNIExtension=false
 -Dsun.rmi.dgc.client.gcInterval=600000 -
Dsun.rmi.dgc.server.gcInterval=600000
```

d) **If the token *REQUIRE_PASSWORD_SECURITY* is enabled, then set a value for the token *PASSWORD_CONTROL_EFFECTIVE_DATE* to set the date from which the Password Control Kit (PCK) begins to disable, delete or expire user accounts. If the effective date is not set, the PCK starts disabling, deleting or expiring user accounts immediately.**

e)

Important: If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

f) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

g) Comment out the following tokens if they already enabled.

```
MAX_WWW_CLIENTS
```

```
PGSQL_MAX_CONNECTIONS
```

```
DATABASE_MAX_POOL_SIZE
```

```
SITE_DIR
```

h) Set the USERS_WITH_NO_EXPIRY_PASSWORD token as follows:

```
USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer
```

i) Configure the database and datamart read-only users settings.

- `DATABASE_READ_ONLY_USER=sitedatabasereadonlyusername`
- `DATABASE_READ_ONLY_PASSWORD=sitedatabasereadonlyuserpwd`
- `REPORTS_DATABASE_READ_ONLY_USER=reportingreadonlyusername`
- `REPORTS_DATABASE_READ_ONLY_PASSWORD=reportingreadonlyuserpwd`

j) Make sure that the following tokens have a value if ETL is enabled.

```
SOAP_ANONYMOUS_SHARED_SECRET=
ETL_SOAP_SHARED_SECRET=
```

k) Configure the following settings for Black Duck Code Sight.

```
HOST_mycodesearchbox.domain.com=codesearch
```

Note: In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>

To enable SSL for Black Duck Code Sight, include this token:
BDCS_SSL=on
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
BDCS_ADMIN_PASSWORD=<password>
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

l) Make sure that the database service names of the production database and reports database are configured to *DATABASE_SERVICE_NAME* and *REPORTS_DATABASE_SERVICE_NAME* respectively.

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

8. **Note:** Execute this step only if 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 this command:

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

9. Run the following command to remove the pagespeed cache.

   ```
   cd /opt/collabnet/teamforge/cache
   rm -rf pagespeed
   ```

10. Log in as the Oracle user and create the read-only database user and permissions.
    a) Create the database read-only user that you will use to connect from CollabNet TeamForge.

       ```
       SQL> create user <database_readonly_user> identified by
       <database_readonly_password> default tablespace <your tablespace >
       temporary tablespace <temporary tablespace>;
       ```

    b) Grant the required permissions to the read-only user that you just created.

       ```
       SQL> grant create session to <database_readonly_user>;
       SQL> exit
       ```

    c) Create the datamart read-only user that you will use to connect from CollabNet TeamForge.

       ```
       SQL> create user <reports_readonly_user> identified by
       <reports_readonly_password> default tablespace <your tablespace>
       temporary tablespace <temporary tablespace>;
       ```

    d) Grant the required permissions to the datamart read-only user that you just created.

       ```
       SQL> grant create session to <reports_readonly_user>;
       SQL> exit
       ```

11. Recreate the runtime environment.

    ```
    cd /opt/collabnet/teamforge-installer/7.0.0.1
    ./install.sh -r -I -V
    ```

12. 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.

13. Convert your site data to work with TeamForge 7.0.

    Tip: Before you kick off the data migration, use the `/etc/init.d/collabnet status` command to make sure the Jboss, Tomcat and Tomcatcs services 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 7.0 site.

14. Run the following post migration script.

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

15. Configure the Apache settings.
    a) Swap in the new Apache configuration file.

       ```
       cd /etc/httpd/conf
       mv httpd.conf httpd.conf_old
       cp httpd.conf.cn_new httpd.conf
       ```

b) Ensure that the Apache configuration file `httpd.conf` is configured with the following settings.

```
<IfModule prefork.c>
StartServers       20
MinSpareServers    10
MaxSpareServers   30
ServerLimit      500
MaxClients       400
MaxRequestsPerChild  4000
</IfModule>
MaxKeepAliveRequests 10000
```

c) Restart Apache.

```
/etc/init.d/httpd restart
```

16. Run the following script to set permissions for the TeamForge database read-only user specified by the *DATABASE_READ_ONLY_USER* token.

```
/opt/collabnet/teamforge/runtime/scripts/set-ctfdb-readonly-user-
permission.py
```

17. Run the following script to set permissions for the reporting database read-only user specified by the REPORTS_DATABASE_READ_ONLY_USER token.

```
/opt/collabnet/teamforge/runtime/scripts/set-reports-readonly-user-
permission.py
```

Do the following to upgrade the SCM box (CVS/Subversion/Git)

18. 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.

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

Directory	Contents
/svnroot	SVN source code repositories.
/cvsroot	CVS source code repositories (not present on all sites).
/gitroot	GIT source code repositories.

```
mkdir -p /tmp/scmbackup_dir
cp -Rpf /svnroot /cvsroot /tmp/scmbackup_dir
```

If GIT integration is enabled, do the following:

```
mkdir /tmp/scmbackup_dir/gerrit
cp –Rpf /gitroot /tmp/scmbackup_dir
cp –Rpf /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
scmbackup_dir/gerrit
cd /tmp
tar czvf 62scmbackup.tgz scmbackup_dir
```

b) Back up your SSH keys, if any.

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

19. Stop TeamForge

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

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

```
mv /etc/yum.repos.d/collabnet-6.2.0.0.repo /etc/yum.repos.d/
collabnet-6.2.0.0.repo.cn_backup
```

21. Set the TeamForge repository configuration.

If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
yum clean all
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) 👉 **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

Insert the Red Hat installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) Mount the Red Hat installation DVD.

```
cd /media/
mkdir cdrom
mount /dev/cdrom ./cdrom/
```

Replace `cdrom` with the identifier for your server's CD drive, if necessary.

👉 **Tip:** If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

22. On the source code box, install the source code component of the TeamForge application.

```
yum clean all
yum install teamforge-scm
```

a) GIT: To install the GIT packages run the following command:

```
yum install teamforge-git
```

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

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

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

a) Use the *SCM_DEFAULT_SHARED_SECRET* on page 382 key from the `site-options.conf` file of the primary TeamForge server

b) Comment out the following token if it is already enabled.

```
MAX_WWW_CLIENTS
```

c) If you're installing GIT, configure the following settings:

```
HOST_myscmbox.host.name=subversion cvs gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

Note: If an older version of TeamForge is integrated with Gerrit and if certain specific properties are customized directly in `gerrit.config>`, the customized properties are not migrated as they were configured when you upgrade to TeamForge 7.0

- Copy the value of teamforge.jumbopushThresold if it is not equal to "30" and add the following property to`site-options.conf` *GERRIT_GIT_PUSH_THRESHOLD* =<your_current_value_of "jumboPushThreshold"_from_gerrit.config>
- Copy the value of teamforge.refreshTimeout if it is not equal to "3600" and add the following property to `site-options.conf` property *GERRIT_GIT_REFRESH_PERIOD* =<your_current_value_of_"refreshTimeout"_from_gerrit.config>

24. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

25. Configure the Apache settings.

a) Swap in the new Apache configuration file.

```
cd /etc/httpd/conf
mv httpd.conf httpd.conf_old
cp httpd.conf.cn_new httpd.conf
```

b) Ensure that the Apache configuration file `httpd.conf` is configured with the following settings.

```
<IfModule prefork.c>
StartServers       20
MinSpareServers    10
MaxSpareServers   30
ServerLimit      500
MaxClients       400
MaxRequestsPerChild  4000
</IfModule>
MaxKeepAliveRequests 10000
```

c) Restart Apache.

```
/etc/init.d/httpd restart
```

26. 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.

27. Start TeamForge

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

28. Start TeamForge.

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

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

29. Integrate Gerrit by running the `post-install.py` script.

```
/opt/collabnet/gerrit/scripts/post-install.py
```

a) Restart the Gerrit services.

```
/etc/init.d/collabnet restart gerrit
```

b) To verify the GIT integration:

Login to the app server and run the following command:

```
/etc/init.d/collabnet status
```

30. If the CLI reports are not configured with the earlier versions of TeamForge, run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

31. If the CLI reports are already configured with the earlier versions of TeamForge, follow these steps:

a) Backup the CLI reports.

```
sudo mkdir /root/backup
cd /root/backup
backup> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> sudo find . -name .svn | xargs sudo rm -rf
```

b) Remove the `cli/reports` folder from the branding repository.

Note: Do a SVN delete to remove this folder from the repository.

```
sudo mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup>sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> sudo svn delete reports
cli> sudo svn commit -m "To delete the old CLI reports folder"
```

c) Run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

d) Commit the back-up 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```
sudo mkdir /root/restore
cd /root/restore
restore> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
sudo mkdir custom-reports
sudo cp -avx /root/backup/pkg custom-reports
sudo svn add custom-reports
sudo svn commit -m "adding the existing customized reports"
```

e) Manually schedule the cron job from the CLI command prompt.

```
/opt/collabnet/teamforge/add-ons/teamforge_cli/bin/ctf
ctf > server add <alise_name> <http/https://hostname>
ctf/list(2)> Now, hit the Enter key.
ctf > conn <alise_name>
```

You may be prompted to enter the username (<admin_username>) and password (<admin_password>)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```

32. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

33. Integrate Black Duck Code Sight with TeamForge by running the Black Duck Code Sight `post-install.sh` script.

Note: It is assumed that Subversion's client configuration file (`/root/.subversion/config`) for the root user is the default one without customization.

Note: If the Black Duck Code Sight is running on a separate server, run the following command in the code sight server.

```
sudo /opt/collabnet/teamforge/runtime/scripts/codesearch/post-install.sh
```

34. After you run the Black Duck Code Sight `post-install.sh` script, run the following script from the same directory:

```
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL similar to "http://myint.box.net/svn/repos" where `myint.box` is the box with the SCM integration server.

Also, if you add a new integration server at some point later, you'll need to run this script, (after creating new integration server), in the TeamForge `app` box.

Important: Do this if you have a multi-box setup with a SCM integration server on a separate server.

35. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.0

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

See *Synchronize TeamForge source control integrations* on page 251.

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 246.

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 239 for details.

f) To verify that the tracker initial load (triggered through `TrackerInitialJob`)is completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

```
select * from etl_job where job_name ='tracker_initial_etl'
```

The status column should have a value of 1.

g) To verify that the SCM initial load (triggered through `SCMCommitInitialJob`)is completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

```
select * from etl_job where job_name ='scm_commit_initial_etl'
```

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

See *Customize anything on your site*.

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

See *Create a site-wide broadcast*.

Important: Do not delete the `teamforge-installer/7.0.0.1` directory. You will need it for future maintenance and upgrades.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Upgrade TeamForge to version 7.0 on new hardware (Oracle)

To upgrade to TeamForge 7.0 without changing the box where your pre TeamForge 7.0 site is running, you have to install the new TeamForge site and migrate the site data.

Important: There are no changes in the steps for installing TeamForge 7.0 Patch 1

In this option, the Oracle database runs on one box, source control services run on the second box and other services run on the application box. 7.0 runs on Red Hat Enterprise Linux/CentOS 6.3/6.4. To take advantage of the new operating system, you should install a fresh box and move your TeamForge 6.2 site to it.

Note: In a multi-box installation of TeamForge, all machines must have the same time zone - without this, ETL will not function properly.

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

1. Back up your site data.

a) Make a dump file of your site database.

To back up the Oracle database, follow the *Oracle backup procedure*.

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

Directory	Contents
/opt/collabnet/teamforge/var	User-created data, such as artifact attachments
/sf-svnroot	Subversion repository for branding data

```
mkdir /tmp/backup_dir
```

```
cp -Rpf /sf-svnroot /opt/collabnet/teamforge/var /tmp/backup_dir
cd /tmp
tar czvf 62backup.tgz backup_dir
```

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

2. Stop the TeamForge application server and the Apache server.

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

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

```
scp /tmp/62backup.tgz username@newbox:/tmp
```

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

```
scp /opt/collabnet/teamforge-installer/6.2.0.0/conf/site-options.conf
username@newbox:/tmp
```

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

Do this on the TeamForge 6.2 source control box

5. 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)
/gitroot	GIT source code repositories.

```
mkdir -p /tmp/scmbackup_dir
cp -Rpf /svnroot /cvsroot /tmp/scmbackup_dir
```

If GIT integration is enabled:

```
mkdir /tmp/scmbackup_dir/gerrit
cp -Rpf /gitroot /tmp/scmbackup_dir
cp -Rpf /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
scmbackup_dir/gerrit
cd /tmp
```

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

6. Stop TeamForge.

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

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

```
sudo scp /opt/collabnet/teamforge-installer/6.2.0.0/conf/site-options.conf
username@newscmbox:/tmp
```

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

```
scp /tmp/scm62backup.tgz username@newscmbox:/tmp
```

Do these steps on the new TeamForge 7.0 application box:

9. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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. Set the TeamForge repository configuration.

    If you are installing TeamForge with internet access:

    - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
    - Copy it to `/etc/yum.repos.d/`.
    - Refresh your repository cache.

      ```
      yum clean all
      ```

    If you are installing TeamForge without internet access:

    a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

    b) Unpack the disconnected installation package.

    ```
    rpm -ivh <package-name>
    ```

    c) **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

    Insert the Red Hat installation DVD.

    This will provide some of the necessary software and utilities for installing TeamForge.

    d) Mount the Red Hat installation DVD.

    ```
    cd /media/
    mkdir cdrom
    mount /dev/cdrom ./cdrom/
    ```

    Replace `cdrom` with the identifier for your server's CD drive, if necessary.

    Tip: If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

    e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

    f) Copy this into the yum configuration file:

    ```
    [RHEL-CDROM]
    name=RHEL CDRom
    baseurl=file:///media/cdrom/Server/
    gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
    enabled=1
    gpgcheck=0
    ```

    g) Run the following commands to check if your yum configuration files are correct.

    ```
    yum list httpd
    yum list apr
    ```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

11. Install the TeamForge application.

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

12. Copy the `site-options.conf` file to the TeamForge installer directory.

```
cp /tmp/site-options.conf /opt/collabnet/teamforge-installer/7.0.0.1/conf
```

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

```
cd /tmp
tar xzvf 62backup.tgz
```

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

```
HOST_my.host.name=app indexer etl
HOST_myscmbox.domain.name=subversion cvs gerrit
HOST_my.db.name=database datamart
DOMAIN_my.host.name=my.domain.name
```

15. In the `site-options.conf` file, make sure you do the following:

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

a) Check the*JAVA_HOME* variable to ensure that TeamForge is using the right JDK.

The JAVA_HOME variable should look like this:

```
JAVA_HOME=/usr/java/jdk1.7.0_21
```

b) If your site is running in SSL mode (SSL=on), add the following java runtime property to the *JBOSS_JAVA_OPTS* token.

```
JBOSS_JAVA_OPTS=-Xms1536m -Xmx1536m -XX:MaxPermSize=512m -server -XX:
+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/tmp -verbose:gc -XX:+
PrintGCTimeStamps -XX:+PrintGCDetails -Djsse.enableSNIExtension=false
 -Dsun.rmi.dgc.client.gcInterval=600000 -
Dsun.rmi.dgc.server.gcInterval=600000
```

c) **If the token *REQUIRE_PASSWORD_SECURITY* is enabled, then set a value for the token *PASSWORD_CONTROL_EFFECTIVE_DATE* to set the date from which the Password Control Kit (PCK) begins to disable, delete or expire user accounts. If the effective date is not set, the PCK starts disabling, deleting or expiring user accounts immediately.**

d)

Important: If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

e) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

f) Comment out the following tokens if they already enabled.

```
MAX_WWW_CLIENTS
```

```
PGSQL_MAX_CONNECTIONS
```

```
DATABASE_MAX_POOL_SIZE
```

```
SITE_DIR
```

g) Set the USERS_WITH_NO_EXPIRY_PASSWORD token as follows:

```
USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer
```

h) Configure the database and datamart read-only users settings.

- `DATABASE_READ_ONLY_USER=sitedatabasereadonlyusername`
- `DATABASE_READ_ONLY_PASSWORD=sitedatabasereadonlyuserpwd`
- `REPORTS_DATABASE_READ_ONLY_USER=reportingreadonlyusername`
- `REPORTS_DATABASE_READ_ONLY_PASSWORD=reportingreadonlyuserpwd`

i) Make sure that the following tokens have a value if ETL is enabled.

```
SOAP_ANONYMOUS_SHARED_SECRET=
ETL_SOAP_SHARED_SECRET=
```

j) Configure the following settings for Black Duck Code Sight.

```
HOST_mycodesearchbox.domain.com=codesearch
```

👉 **Note:** In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>

To enable SSL for Black Duck Code Sight, include this token:
BDCS_SSL=on
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
BDCS_ADMIN_PASSWORD=<password>
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

k) Make sure that the database service names of the production database and reports database are configured to *DATABASE_SERVICE_NAME* and *REPORTS_DATABASE_SERVICE_NAME* respectively.

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

16. 👉 **Note:** Execute this step only if 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 this command:

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

17. Log in as the Oracle user and create the read-only database user and permissions.
    a) Create the database read-only user that you will use to connect from CollabNet TeamForge.

    ```
    SQL> create user <database_readonly_user> identified by
    <database_readonly_password> default tablespace <your tablespace >
    temporary tablespace <temporary tablespace>;
    ```

    b) Grant the required permissions to the read-only user that you just created.

    ```
    SQL> grant create session to <database_readonly_user>;
    SQL> exit
    ```

    c) Create the datamart read-only user that you will use to connect from CollabNet TeamForge.

    ```
    SQL> create user <reports_readonly_user> identified by
    <reports_readonly_password> default tablespace <your tablespace>
    temporary tablespace <temporary tablespace>;
    ```

    d) Grant the required permissions to the datamart read-only user that you just created.

    ```
    SQL> grant create session to <reports_readonly_user>;
    SQL> exit
    ```

18. Recreate the runtime environment.

    ```
    cd /opt/collabnet/teamforge-installer/7.0.0.1
    ./install.sh -r -I -V
    ```

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 7.0.

    Tip: Before you kick off the data migration, use the `/etc/init.d/collabnet status` command to make sure the Jboss, Tomcat and Tomcatcs services 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 7.0 site.

21. Run the following post migration script.

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

22. Configure the Apache settings.
    a) Swap in the new Apache configuration file.

    ```
    cd /etc/httpd/conf
    mv httpd.conf httpd.conf_old
    cp httpd.conf.cn_new httpd.conf
    ```

    b) Ensure that the Apache configuration file `httpd.conf` is configured with the following settings.

    ```
    <IfModule prefork.c>
    StartServers       20
    MinSpareServers    10
    MaxSpareServers   30
    ServerLimit      500
    ```

```
MaxClients       400
MaxRequestsPerChild  4000
</IfModule>
MaxKeepAliveRequests 10000
```

c) Restart Apache.

```
/etc/init.d/httpd restart
```

Do this on the new TeamForge 7.0 source control box

23. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.

- The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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.

24. Check your basic networking setup.

See *Set up networking for your TeamForge box* on page 7 for details.

25. Set the TeamForge repository configuration.

If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
yum clean all
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

Insert the Red Hat installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) Mount the Red Hat installation DVD.

```
cd /media/
mkdir cdrom
mount /dev/cdrom ./cdrom/
```

Replace `cdrom` with the identifier for your server's CD drive, if necessary.

Tip: If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

26. On the source code box, install the source code component of the TeamForge application.

```
yum clean all
yum install teamforge-scm
```

a) GIT: To install the GIT packages run the following command:

```
yum install teamforge-git
```

27. Copy the `site-options.conf` file to the TeamForge installer directory.

```
cp /tmp/site-options.conf /opt/collabnet/teamforge-installer/7.0.0.1/conf
```

28. Reload svnroot, cvsroot and gitroot.

```
tar zxvf /tmp/62scmbackup.tgz
cp -Rpf /tmp/scmbackup_dir/svnroot /svnroot
cp -Rpf /tmp/scmbackup_dir/cvsroot /cvsroot
```

If GIT integration is enabled, do the following:

```
cp -Rpf /tmp/scmbackup_dir/gitroot /gitroot
cp -Rpf /tmp/scmbackup_dir/gerrit/etc /opt/collabnet/gerrit/
cp -Rpf /tmp/scmbackup_dir/gerrit/.ssh /opt/collabnet/gerrit/
```

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

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

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

a) Use the *SCM_DEFAULT_SHARED_SECRET* on page 382 key from the `site-options.conf` file of the primary TeamForge server

b) Increase the value of MAX_WWW_CLIENTS.

```
MAX_WWW_CLIENTS=220
```

c) If you're installing GIT, configure the following settings:

```
HOST_my.host.name=app etl database datamart subversion cvs
 indexer gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

> **Note:** If an older version of TeamForge is integrated with Gerrit and if certain specific properties are customized directly in `gerrit.config>`, the customized properties are not migrated as they were configured when you upgrade to TeamForge 7.0

- Copy the value of teamforge.jumbopushThresold if it is not equal to "30" and add the following property to`site-options.conf` *GERRIT_GIT_PUSH_THRESHOLD* =<your_current_value_of "jumboPushThreshold"_from_gerrit.config>
- Copy the value of teamforge.refreshTimeout if it is not equal to "3600" and add the following property to `site-options.conf` property *GERRIT_GIT_REFRESH_PERIOD* =<your_current_value_of_"refreshTimeout"_from_gerrit.config>

30. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

31. 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.

32. Start TeamForge

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

33. Start TeamForge.

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

> **Note:** The sample TeamForge project template is installed as part of the collabnetstartup script. If the project template is installed, or it exists in the database by default, the startup script will not re-install it.

34. Integrate Gerrit by running the `post-install.py` script.

```
/opt/collabnet/gerrit/scripts/post-install.py
```

a) Restart the Gerrit services.

```
/etc/init.d/collabnet restart gerrit
```

b) To verify the GIT integration:

Login to the app server and run the following command:

```
/etc/init.d/collabnet status
```

35. If the CLI reports are not configured with the earlier versions of TeamForge, run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

36. If the CLI reports are already configured with the earlier versions of TeamForge, follow these steps:

a) Backup the CLI reports.

```
sudo mkdir /root/backup
cd /root/backup
backup> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> sudo find . -name .svn | xargs sudo rm -rf
```

b) Remove the `cli/reports` folder from the branding repository.

> **Note:** Do a SVN delete to remove this folder from the repository.

```
sudo mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup>sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
```

```
cd cli
cli> sudo svn delete reports
cli> sudo svn commit -m "To delete the old CLI reports folder"
```

c) Run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

d) Commit the back-up 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```
sudo mkdir /root/restore
cd /root/restore
restore> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
sudo mkdir custom-reports
sudo cp -avx /root/backup/pkg custom-reports
sudo svn add custom-reports
sudo svn commit -m "adding the existing customized reports"
```

e) Manually schedule the cron job from the CLI command prompt.

```
/opt/collabnet/teamforge/add-ons/teamforge_cli/bin/ctf
ctf > server add <alise_name> <http/https://hostname>
ctf/list(2)> Now, hit the Enter key.
ctf > conn <alise_name>
```

You may be prompted to enter the username (<admin_username>) and password (<admin_password>)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```

37. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

38. Integrate Black Duck Code Sight with TeamForge by running the Black Duck Code Sight `post-install.sh` script.

Note: It is assumed that Subversion's client configuration file (`/root/.subversion/config`) for the root user is the default one without customization.

Note: If the Black Duck Code Sight is running on a separate server, run the following command in the code sight server.

```
sudo /opt/collabnet/teamforge/runtime/scripts/codesearch/post-install.sh
```

39. After you run the Black Duck Code Sight `post-install.sh` script, run the following script from the same directory:

```
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL similar to "http://myint.box.net/svn/repos" where `myint.box` is the box with the SCM integration server.

Also, if you add a new integration server at some point later, you'll need to run this script, (after creating new integration server), in the TeamForge `app` box.

Important: Do this if you have a multi-box setup with a SCM integration server on a separate server.

40. If the token *VALIDATE_SSL_CERTS* is set to "true", you must run the codesearch runtime script *trust-cert.sh* in the application box.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
```

41. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.0
    a) Make sure your users can still access their source control services.

       See *Synchronize TeamForge source control integrations* on page 251.
    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 246.
    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 239 for details.
    f) To verify that the tracker initial load (triggered through `TrackerInitialJob`)is completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

       ```
       select * from etl_job where job_name ='tracker_initial_etl'
       ```

       The status column should have a value of 1.
    g) To verify that the SCM initial load (triggered through `SCMCommitInitialJob`)is completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

       ```
       select * from etl_job where job_name ='scm_commit_initial_etl'
       ```
    h) If your site has custom branding, verify that your branding changes still work as intended.

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

       See *Create a site-wide broadcast*.

Important: Do not delete the `teamforge-installer/7.0.0.1` directory. You will need it for future maintenance and upgrades.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Upgrade to TeamForge 7.0 on SuSE

You can upgrade to TeamForge 7.0 from TeamForge 6.2. 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: The zypper installer will install the CollabNet TeamForge site in the default directory "/opt/collabnet/teamforge" only.

Note: Due to the index content change for the search functionality in TeamForge 6.1.1, a complete re-indexing of the site is required if you are upgrading from TeamForge 6.1 or earlier versions of TeamForge. This could take several hours and the index data could double in size.

Upgrade to TeamForge 7.0 on the same box - Black Duck Code Sight on a separate box

To upgrade to TeamForge 7.0, you install the new site and convert your existing site's data.

Important: There are no changes in the steps to upgrade to TeamForge 7.0 Patch 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. However, the priority here is to keep things as simple and quick as possible as we are working with a dedicated installation. See *Should I upgrade to TeamForge 7.0 on a new box?* on page 290 for some background information.

Tip: If there is any doubt about what kind of site you are working with, see *Is my TeamForge site "dedicate"d or "advanced"?* on page 223

Note: In a multi-box installation of TeamForge, all machines must have the same time zone - without this, ETL will not function properly.

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/9.0/backups/
      teamforge_data_backup.dmp
      exit
      mkdir /tmp/backup_dir
      cp /var/lib/pgsql/9.0/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/9.0/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 62backup.tgz backup_dir
      ```

   c) Back up your SSH keys, if any.
   d) Back up your SSL certificates and keys, if any.
   e) If you have installed GIT integration, back up the `gerrit.config` file.

      ```
      cd /opt/collabnet/gerrit/etc
      cp gerrit.config gerrit.config.b4upg
      ```

2. Stop TeamForge.

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

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

   ```
   mv /etc/zypp/repos.d/collabnet-6.2.0.0.repo /etc/zypp/repos.d/
   collabnet-6.2.0.0.repo.cn_backup
   ```

4. If you are installing TeamForge with internet access:

   - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
   - Copy it to `/etc/yum.repos.d/`.
   - Refresh your repository cache.

     ```
     zypper ref
     ```

   If you are installing TeamForge without internet access:

   a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

   b) Unpack the disconnected installation package.

      ```
      rpm -ivh <package-name>
      ```

   c) **Note:** If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

      Insert the SuSE installation DVD.

      This will provide some of the necessary software and utilities for installing TeamForge.

   d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

      If necessary, replace `sr0` with the identifier for your server's CD drive.

   Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.

5. Install the following application packages.

   a) TeamForge: To install the TeamForge application packages run the following command:

      ```
      zypper install teamforge
      ```

   b) GIT: To install the GIT packages run the following command:

      ```
      zypper install teamforge-git
      ```

6. In the `site-options.conf` file, make sure you do the following:

   Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

   a) Update the host name and domain name, if required.

      HOST_my.host.name=app database subversion cvs etl datamart `indexer`

b) If you're installing GIT, configure the following settings:

```
HOST_my.host.name=app etl database datamart subversion cvs gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

Note: If an older version of TeamForge is integrated with Gerrit and if certain specific properties are customized directly in `gerrit.config>`, the customized properties are not migrated as they were configured when you upgrade to TeamForge 7.0

- Copy the value of teamforge.jumbopushThresold if it is not equal to "30" and add the following property to`site-options.conf` *GERRIT_GIT_PUSH_THRESHOLD* =<your_current_value_of "jumboPushThreshold"_from_gerrit.config>
- Copy the value of teamforge.refreshTimeout if it is not equal to "3600" and add the following property to `site-options.conf` property *GERRIT_GIT_REFRESH_PERIOD* =<your_current_value_of_"refreshTimeout"_from_gerrit.config>

c) Check the*JAVA_HOME* variable to ensure that TeamForge is using the right JDK.

The JAVA_HOME variable should look like this:

```
JAVA_HOME=/usr/java/jdk1.7.0_21
```

d) If your site is running in SSL mode (SSL=on), add the following java runtime property to the *JBOSS_JAVA_OPTS* token.

```
JBOSS_JAVA_OPTS=-Xms1536m -Xmx1536m -XX:MaxPermSize=512m -server -XX:
+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/tmp -verbose:gc -XX:+
PrintGCTimeStamps -XX:+PrintGCDetails -Djsse.enableSNIExtension=false
 -Dsun.rmi.dgc.client.gcInterval=600000 -
Dsun.rmi.dgc.server.gcInterval=600000
```

e) **If the token *REQUIRE_PASSWORD_SECURITY* is enabled, then set a value for the token *PASSWORD_CONTROL_EFFECTIVE_DATE* to set the date from which the Password Control Kit (PCK) begins to disable, delete or expire user accounts. If the effective date is not set, the PCK starts disabling, deleting or expiring user accounts immediately.**

f)

Important: If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

g) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

h) Comment out the following tokens if they already enabled.

```
MAX_WWW_CLIENTS
```

```
PGSQL_MAX_CONNECTIONS
```

```
DATABASE_MAX_POOL_SIZE
```

```
SITE_DIR
```

i) Set the USERS_WITH_NO_EXPIRY_PASSWORD token as follows:

```
USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer
```

j) Make sure that the following tokens have a value if ETL is enabled.

```
SOAP_ANONYMOUS_SHARED_SECRET=
ETL_SOAP_SHARED_SECRET=
```

k) Configure the following settings for Black Duck Code Sight.

```
HOST_mycodesearchbox.domain.com=app database subversion cvs etl datamart
 indexer gerrit codesearch
```

Note: In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>

To enable SSL for Black Duck Code Sight, include this token:
BDCS_SSL=on
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
BDCS_ADMIN_PASSWORD=<password>
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

7. Run the following command to remove the pagespeed cache.

```
cd /opt/collabnet/teamforge/cache
rm -rf pagespeed
```

8. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

9. 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.

Do this on the Black Duck Code Sight server. We'll call this `my.code.box`.

10. Set up the Black Duck Code Sight server using *these instructions*.

Do this on my.app.box:

11. Convert your site data to work with TeamForge 7.0.

Tip: Before you kick off the data migration, use the `/etc/init.d/collabnet status` command to make sure the Jboss, Tomcat and Tomcatcs services 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 7.0 site.

12. Run the following post migration script.

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

13. Run the Subversion working copy script when you are upgrading to TeamForge 6.2 and the old Teamforge site has the SVN 1.6.x version.
    a) `cd <SITE_DIR>/runtime/scripts`
    b) `./svn-upgrade-working-copies.sh`

    Note: Type "Y" when prompted.

14. Start TeamForge.

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

    Note:

    - The TeamForge startup script installs the sample TeamForge project templates by default.
    - If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
    - You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

15. If the CLI reports are not configured with the earlier versions of TeamForge, run the `post-install.py` script.

    ```
    sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
    ```

16. If the CLI reports are already configured with the earlier versions of TeamForge, follow these steps:
    a) Backup the CLI reports.

    ```
    sudo mkdir /root/backup
    cd /root/backup
    backup> sudo svn checkout --username XXXXX https://<hostname>/svn/
    repository-internal/branding/cli/reports/pkg
    backup> sudo find . -name .svn | xargs sudo rm -rf
    ```

    b) Remove the `cli/reports` folder from the branding repository.

    Note: Do a SVN delete to remove this folder from the repository.

    ```
    sudo mkdir /tmp/cleanup
    cd /tmp/cleanup
    cleanup>sudo svn checkout --username XXXXX https://<hostname>/svn/
    repository-internal/branding/cli/
    cd cli
    cli> sudo svn delete reports
    cli> sudo svn commit -m "To delete the old CLI reports folder"
    ```

    c) Run the `post-install.py` script.

    ```
    sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
    ```

    d) Commit the back-up 'pkg' folder which is available in this location `branding/cli/custom-reports/`

    ```
    sudo mkdir /root/restore
    cd /root/restore
    restore> sudo svn checkout --username XXXXX https://<hostname>/svn/
    repository-internal/branding/cli/
    cd cli
    sudo mkdir custom-reports
    sudo cp -avx /root/backup/pkg custom-reports
    sudo svn add custom-reports
    sudo svn commit -m "adding the existing customized reports"
    ```

    e) Manually schedule the cron job from the CLI command prompt.

```
/opt/collabnet/teamforge/add-ons/teamforge_cli/bin/ctf
ctf > server add <alise_name> <http/https://hostname>
ctf/list(2)> Now, hit the Enter key.
ctf > conn <alise_name>
```

You may be prompted to enter the username (<admin_username>) and password (<admin_password>)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```

17. Integrate gerrit by running the `post-install.py` script.

Tip: Skip this step if you are not installing git integration.

```
/opt/collabnet/gerrit/scripts/post-install.py
```

The post installation script will try to detect the required configuration parameters. The following 3 parameters do not have default values and you will be asked to provide them:

- Teamforge Login name: the dedicated TeamForge site administrator account that does not expire and cannot be locked
- Teamforge Password: the password for the above account
- Database password: the password to protect Gerrit's database from unauthorized access. Specify its value when you first run the post-install script. Make sure you note the value because you will be asked for it later.

a) Restart the Gerrit services.

```
/etc/init.d/collabnet restart gerrit
```

b) To verify the Git integration, run the following command.

```
/etc/init.d/collabnet status
```

18. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

19. Integrate Black Duck Code Sight with TeamForge by running the Black Duck Code Sight `post-install.sh` script.

Note: It is assumed that Subversion's client configuration file (`/root/.subversion/config`) for the root user is the default one without customization.

Note: If the Black Duck Code Sight is running on a separate server, run the following command in the code sight server.

```
sudo /opt/collabnet/teamforge/runtime/scripts/codesearch/post-install.sh
```

20. After you run the Black Duck Code Sight `post-install.sh` script, run the following script from the same directory:

```
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL similar to "http://myint.box.net/svn/repos" where `myint.box` is the box with the SCM integration server.

Also, if you add a new integration server at some point later, you'll need to run this script, (after creating new integration server), in the TeamForge `app` box.

Important: Do this if you have a multi-box setup with a SCM integration server on a separate server.

21. If the token *VALIDATE_SSL_CERTS* is set to "true", you must run the codesearch runtime script *trust-cert.sh* in the application box.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
```

22. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.0.
    a) Make sure your users can still access their source control services.

       See *Synchronize TeamForge source control integrations* on page 251.
    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 246.
    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 239 for details.
    f) To verify that the tracker initial load (triggered through `TrackerInitialJob`) completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

       ```
       select * from etl_job where job_name ='tracker_initial_etl'
       ```

       The status column should have a value of 1.
    g) If your site has custom branding, verify that your branding changes still work as intended.

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

       See *Create a site-wide broadcast*.

Important: Do not delete the `teamforge-installer/7.0.0.1` 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.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Upgrade to TeamForge 7.0 on new hardware - Black Duck Code Sight on same box

To upgrade to TeamForge 7.0, set up a new box, then bring over your old site's data and convert it.

Important: There are no changes in the steps to upgrade to TeamForge 7.0 Patch 1

TeamForge 7.0 runs on SLES 11 SP2. To take advantage of the new operating system, you should install a fresh box and move your TeamForge site to it.

Note: In a multi-box installation of TeamForge, all machines must have the same time zone - without this, ETL will not function properly.

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

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/9.0/backups/
teamforge_data_backup.dmp
exit
mkdir /tmp/backup_dir
cp /var/lib/pgsql/9.0/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/9.0/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 62backup.tgz backup_dir
```

c) Back up your SSH keys, if any.

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

e) If you have installed GIT integration, back up the `gerrit.config` file.

```
cd /opt/collabnet/gerrit/etc
cp gerrit.config gerrit.config.b4upg
```

2. Stop the TeamForge application server and the Apache server.

```
/etc/init.d/apache2 stop
/etc/init.d/collabnet stop
```

3. Stop PostgreSQL.

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

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

```
scp /opt/collabnet/teamforge-installer/6.2.0.0/conf/site-options.conf
username@newbox:/tmp
```

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

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

```
scp /tmp/62backup.tgz username@newbox:/tmp
```

6. If the old server has the TeamForge CLI server add-on installed, then copy the cliserver.properties file under `/etc/` from the old server to`/shared/cliserver`

Note: Skip this step if you are migrating from TeamForge 6.2.0.1

```
sudo cp /etc/cliserver.properties /shared/teamforge/etc
```

Do this on the new TeamForge 7.0 box:

7. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.
   - See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
   - See *the SuSE Linux Enterprise Server deployment guide* for help.

   Important: Don't customize your installation. Select only the default packages list.
8. Check your basic networking setup.

   See *Set up networking for your TeamForge box* on page 7 for details.
9. If you are installing TeamForge with internet access:
   - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
   - Copy it to `/etc/yum.repos.d/`.
   - Refresh your repository cache.

     ```
     zypper ref
     ```

   If you are installing TeamForge without internet access:

   a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

   b) Unpack the disconnected installation package.

   ```
   rpm -ivh <package-name>
   ```

   c) **Note:** If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

   Insert the SuSE installation DVD.

   This will provide some of the necessary software and utilities for installing TeamForge.

   d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

   If necessary, replace `sr0` with the identifier for your server's CD drive.

   Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.
10. Install the following application packages.

    a) TeamForge: To install the TeamForge application packages run the following command:

    ```
    zypper install teamforge
    ```

    b) Black Duck Code Sight: To install Black Duck Code Sight run the following command:

    ```
    zypper install teamforge-codesearch
    ```

    c) GIT: To install the GIT packages run the following command:

    ```
    zypper install teamforge-git
    ```
11. Copy the `site-options.conf` file to the TeamForge installer directory.

```
cp /tmp/site-options.conf /opt/collabnet/teamforge-installer/7.0.0.1/conf
```

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

```
cd /tmp
tar xzvf 62backup.tgz
```

13. In the `site-options.conf` file, make sure you do the following:

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

a) Update the host name and domain name, if required.

HOST_my.host.name=app database subversion cvs etl datamart `indexer`

b) If you're installing GIT, configure the following settings:

```
HOST_my.host.name=app etl database datamart subversion cvs gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

Note: If an older version of TeamForge is integrated with Gerrit and if certain specific properties are customized directly in `gerrit.config>`, the customized properties are not migrated as they were configured when you upgrade to TeamForge 7.0

- Copy the value of teamforge.jumbopushThresold if it is not equal to "30" and add the following property to`site-options.conf` *GERRIT_GIT_PUSH_THRESHOLD* =<your_current_value_of "jumboPushThreshold"_from_gerrit.config>
- Copy the value of teamforge.refreshTimeout if it is not equal to "3600" and add the following property to `site-options.conf` property *GERRIT_GIT_REFRESH_PERIOD* =<your_current_value_of_"refreshTimeout"_from_gerrit.config>

c) Check the*JAVA_HOME* variable to ensure that TeamForge is using the right JDK.

The JAVA_HOME variable should look like this:

```
JAVA_HOME=/usr/java/jdk1.7.0_21
```

d) If your site is running in SSL mode (SSL=on), add the following java runtime property to the *JBOSS_JAVA_OPTS* token.

```
JBOSS_JAVA_OPTS=-Xms1536m -Xmx1536m -XX:MaxPermSize=512m -server -XX:
+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/tmp -verbose:gc -XX:+
PrintGCTimeStamps -XX:+PrintGCDetails -Djsse.enableSNIExtension=false
 -Dsun.rmi.dgc.client.gcInterval=600000 -
Dsun.rmi.dgc.server.gcInterval=600000
```

e) **If the token *REQUIRE_PASSWORD_SECURITY* is enabled, then set a value for the token *PASSWORD_CONTROL_EFFECTIVE_DATE* to set the date from which the Password Control Kit (PCK) begins to disable, delete or expire user accounts. If the effective date is not set, the PCK starts disabling, deleting or expiring user accounts immediately.**

f)

Important: If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

g) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

h) Add the token *DEDICATED_INSTALL=true* if it is not present in the `site-options.conf` file.

The dedicated installation mode is suitable when the server is completely dedicated to TeamForge application. This makes the installation process very simple as theTeamForge installer takes care of configuring your apache, postgresql automatically without any manual intervention. In a "Dedicated" install, it is advised not to install another software application other than TeamForge. It is also recommended not to run any unwanted services that are not used by TeamForge.

i) Comment out the following tokens if they already enabled.

```
MAX_WWW_CLIENTS
```

```
PGSQL_MAX_CONNECTIONS
```

```
DATABASE_MAX_POOL_SIZE
```

```
SITE_DIR
```

j) Set the USERS_WITH_NO_EXPIRY_PASSWORD token as follows:

```
USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer
```

k) Make sure that the following tokens have a value if ETL is enabled.

```
SOAP_ANONYMOUS_SHARED_SECRET=
ETL_SOAP_SHARED_SECRET=
```

l) Configure the following settings for Black Duck Code Sight.

```
HOST_mycodesearchbox.domain.com=app database subversion cvs etl datamart
 indexer gerrit codesearch
```

Note: In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>

To enable SSL for Black Duck Code Sight, include this token:
BDCS_SSL=on
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
BDCS_ADMIN_PASSWORD=<password>
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

14. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

15. 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`, `gitroot` and `var`.

```
tar zxvf /tmp/62backup.tgz
cp -Rpf /tmp/backup_dir/svnroot /svnroot
cp -Rpf /tmp/backup_dir/cvsroot /cvsroot
cp -Rpf /tmp/backup_dir/sf-svnroot /sf-svnroot
cp -Rpf /tmp/backup_dir/var /opt/collabnet/teamforge/var
```

If GIT integration is enabled:

```
cp -Rpf /tmp/backup_dir/gitroot /gitroot
cp -Rpf /tmp/backup_dir/gerrit/etc /opt/collabnet/gerrit/
cp -Rpf /tmp/backup_dir/gerrit/.ssh /opt/collabnet/gerrit/
```

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.

Do this on my.app.box:

17. Convert your site data to work with TeamForge 7.0.

Tip: Before you kick off the data migration, use the `/etc/init.d/collabnet status` command to make sure the Jboss, Tomcat and Tomcatcs services 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 7.0 site.

18. Run the following post migration script.

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

19. Start TeamForge.

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

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

20. If the CLI reports are not configured with the earlier versions of TeamForge, run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

21. If the CLI reports are already configured with the earlier versions of TeamForge, follow these steps:

a) Backup the CLI reports.

```
sudo mkdir /root/backup
cd /root/backup
backup> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
```

```
backup> sudo find . -name .svn | xargs sudo rm -rf
```

b) Remove the `cli/reports` folder from the branding repository.

👉 **Note:** Do a SVN delete to remove this folder from the repository.

```
sudo mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup>sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> sudo svn delete reports
cli> sudo svn commit -m "To delete the old CLI reports folder"
```

c) Run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

d) Commit the back-up 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```
sudo mkdir /root/restore
cd /root/restore
restore> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
sudo mkdir custom-reports
sudo cp -avx /root/backup/pkg custom-reports
sudo svn add custom-reports
sudo svn commit -m "adding the existing customized reports"
```

e) Manually schedule the cron job from the CLI command prompt.

```
/opt/collabnet/teamforge/add-ons/teamforge_cli/bin/ctf
ctf > server add <alise_name> <http/https://hostname>
ctf/list(2)> Now, hit the Enter key.
ctf > conn <alise_name>
```

You may be prompted to enter the username (<admin_username>) and password (<admin_password>)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```

22. Integrate Gerrit by running the `post-install.py` script.

```
/opt/collabnet/gerrit/scripts/post-install.py
```

a) Restart the Gerrit services.

```
/etc/init.d/collabnet restart gerrit
```

b) To verify the GIT integration:

Login to the app server and run the following command:

```
/etc/init.d/collabnet status
```

23. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

24. Integrate Black Duck Code Sight with TeamForge by running the Black Duck Code Sight `post-install.sh` script.

Note: It is assumed that Subversion's client configuration file (`/root/.subversion/config`) for the root user is the default one without customization.

Note: If the Black Duck Code Sight is running on a separate server, run the following command in the code sight server.

```
sudo /opt/collabnet/teamforge/runtime/scripts/codesearch/post-install.sh
```

25. After you run the Black Duck Code Sight `post-install.sh` script, run the following script from the same directory:

```
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL similar to "http://myint.box.net/svn/repos" where `myint.box` is the box with the SCM integration server.

Also, if you add a new integration server at some point later, you'll need to run this script, (after creating new integration server), in the TeamForge `app` box.

Important: Do this if you have a multi-box setup with a SCM integration server on a separate server.

26. If the token *VALIDATE_SSL_CERTS* is set to "true", you must run the codesearch runtime script *trust-cert.sh* in the application box.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
```

27. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.0

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

See *Synchronize TeamForge source control integrations* on page 251.

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 246.

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 239 for details.

f) To verify that the tracker initial load (triggered through `TrackerInitialJob`)is completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

```
select * from etl_job where job_name ='tracker_initial_etl'
```

The status column should have a value of 1.

g) To verify that the SCM initial load (triggered through `SCMCommitInitialJob`)is completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

```
select * from etl_job where job_name ='scm_commit_initial_etl'
```

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

See *Customize anything on your site*.

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

See *Create a site-wide broadcast*.

Important: Do not delete the `teamforge-installer/7.0.0.1` 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.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Upgrade to TeamForge 7.0 on the same hardware - SCM on a separate box

To upgrade to TeamForge 7.0, you install the new site and convert your existing site's data.

Important: There are no changes in the steps to upgrade to TeamForge 7.0 Patch 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, 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 7.0 on a new box?* on page 290 for some background information.

Tip: If there is any doubt about what kind of site you are working with, see *Is my TeamForge site "dedicate"d or "advanced"?* on page 223

Note: In a multi-box installation of TeamForge, all machines must have the same time zone - without this, ETL will not function properly.

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/9.0/backups/
      teamforge_data_backup.dmp
      exit
      mkdir /tmp/backup_dir
      cp /var/lib/pgsql/9.0/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/9.0/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
/sf-svnroot	Subversion repository for branding data

      ```
      cp -Rpf /sf-svnroot /opt/collabnet/teamforge/var /tmp/backup_dir
      cd /tmp
      tar czvf 62backup.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. Move the repositories from any previous installs out of the way.

   ```
   mv /etc/zypp/repos.d/collabnet-6.2.0.0.repo /etc/zypp/repos.d/
   collabnet-6.2.0.0.repo.cn_backup
   ```

4. If you are installing TeamForge with internet access:

   - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
   - Copy it to `/etc/yum.repos.d/`.
   - Refresh your repository cache.

     ```
     zypper ref
     ```

   If you are installing TeamForge without internet access:

   a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

   b) Unpack the disconnected installation package.

      ```
      rpm -ivh <package-name>
      ```

   c) **Note:** If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

      Insert the SuSE installation DVD.

      This will provide some of the necessary software and utilities for installing TeamForge.

   d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

      If necessary, replace `sr0` with the identifier for your server's CD drive.

   Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.

5. Install the following application packages.

   a) TeamForge: To install the TeamForge application packages run the following command:

      ```
      zypper install teamforge-app teamforge-etl teamforge-database
      ```

6. In the `site-options.conf` file, make sure you do the following:

   Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

   a) Update the host name and domain name, if required.

      HOST_my.host.name=app etl database datamart `indexer`
      HOST_myscmbox.domain.name=subversion cvs

   b) Check the*JAVA_HOME* variable to ensure that TeamForge is using the right JDK.

The JAVA_HOME variable should look like this:

```
JAVA_HOME=/usr/java/jdk1.7.0_21
```

c) If your site is running in SSL mode (SSL=on), add the following java runtime property to the *JBOSS_JAVA_OPTS* token.

```
JBOSS_JAVA_OPTS=-Xms1536m -Xmx1536m -XX:MaxPermSize=512m -server -XX:
+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/tmp -verbose:gc -XX:+
PrintGCTimeStamps -XX:+PrintGCDetails -Djsse.enableSNIExtension=false
 -Dsun.rmi.dgc.client.gcInterval=600000 -
Dsun.rmi.dgc.server.gcInterval=600000
```

d) **If the token *REQUIRE_PASSWORD_SECURITY* is enabled, then set a value for the token *PASSWORD_CONTROL_EFFECTIVE_DATE* to set the date from which the Password Control Kit (PCK) begins to disable, delete or expire user accounts. If the effective date is not set, the PCK starts disabling, deleting or expiring user accounts immediately.**

e) 👉 **Important:** If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

f) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

g) Add the token *DEDICATED_INSTALL=true*

This makes the installation process very simple as theTeamForge installer takes care of configuring the apache, postgresql automatically without any manual intervention.

h) Comment out the following tokens if they already enabled.

```
MAX_WWW_CLIENTS
```

```
PGSQL_MAX_CONNECTIONS
```

```
DATABASE_MAX_POOL_SIZE
```

```
SITE_DIR
```

i) Set the REPORTS_DATABASE_MAX_POOL_SIZE token as follows:

```
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

j) Increase the ETL heap size.

```
ETL_JAVA_OPTS=-Xms160m -Xmx512m -server -XX:+HeapDumpOnOutOfMemoryError
  -XX:HeapDumpPath=/tmp -verbose:gc -XX:+PrintGCTimeStamps -
XX:+PrintGCDetails -Dsun.rmi.dgc.client.gcInterval=600000 -
Dsun.rmi.dgc.server.gcInterval=600000
```

k) Set the USERS_WITH_NO_EXPIRY_PASSWORD token as follows:

```
USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer
```

l) Make sure that the following tokens have a value if ETL is enabled.

```
SOAP_ANONYMOUS_SHARED_SECRET=
ETL_SOAP_SHARED_SECRET=
```

m) Configure the following settings for Black Duck Code Sight.

```
HOST_mycodesearchbox.domain.com=codesearch
```

Note: In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>

To enable SSL for Black Duck Code Sight, include this token:
BDCS_SSL=on
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
BDCS_ADMIN_PASSWORD=<password>
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

7. Run the following command to remove the pagespeed cache.

```
cd /opt/collabnet/teamforge/cache
rm -rf pagespeed
```

8. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

9. 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.

10. Convert your site data to work with TeamForge 7.0.

Tip: Before you kick off the data migration, use the `/etc/init.d/collabnet status` command to make sure the Jboss, Tomcat and Tomcatcs services 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 7.0 site.

11. Run the following post migration script.

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

12. Start TeamForge.

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

13. If the CLI reports are not configured with the earlier versions of TeamForge, run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

14. If the CLI reports are already configured with the earlier versions of TeamForge, follow these steps:

a) Backup the CLI reports.

```
sudo mkdir /root/backup
```

```
cd /root/backup
backup> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> sudo find . -name .svn | xargs sudo rm -rf
```

b) Remove the `cli/reports` folder from the branding repository.

Note: Do a SVN delete to remove this folder from the repository.

```
sudo mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup>sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> sudo svn delete reports
cli> sudo svn commit -m "To delete the old CLI reports folder"
```

c) Run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

d) Commit the back-up 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```
sudo mkdir /root/restore
cd /root/restore
restore> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
sudo mkdir custom-reports
sudo cp -avx /root/backup/pkg custom-reports
sudo svn add custom-reports
sudo svn commit -m "adding the existing customized reports"
```

e) Manually schedule the cron job from the CLI command prompt.

```
/opt/collabnet/teamforge/add-ons/teamforge_cli/bin/ctf
ctf > server add <alise_name> <http/https://hostname>
ctf/list(2)> Now, hit the Enter key.
ctf > conn <alise_name>
```

You may be prompted to enter the username (<admin_username>) and password (<admin_password>)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```

15. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

16. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.0

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

See *Synchronize TeamForge source control integrations* on page 251.

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 246.

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 239 for details.

f) To verify that the tracker initial load (triggered through `TrackerInitialJob`)is completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

```
select * from etl_job where job_name ='tracker_initial_etl'
```

The status column should have a value of 1.

g) To verify that the SCM initial load (triggered through `SCMCommitInitialJob`)is completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

```
select * from etl_job where job_name ='scm_commit_initial_etl'
```

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

See *Customize anything on your site*.

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

See *Create a site-wide broadcast*.

Important: Do not delete the `teamforge-installer/7.0.0.1` directory. You will need it for future maintenance and upgrades.

Do the following to upgrade the SCM box (CVS/Subversion/Git)

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.

17. 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)
opt/collabnet/gerrit/etc	opt/collabnet/gerrit/etc
/opt/collabnet/gerrit/.ssh	Gerrit SSH Keys

```
mkdir -p /tmp/scmbackup_dir/gerrit
cp -Rpf /svnroot /cvsroot /tmp/scmbackup_dir
cp -Rpf /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
scmbackup_dir/gerrit
cd /tmp
tar czvf 62scmbackup.tgz scmbackup_dir
```

a) Back up your SSH keys, if any.

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

18. Stop TeamForge

```
/etc/init.d/apache2 stop
```

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

```
mv /etc/zypp/repos.d/collabnet-6.2.0.0.repo /etc/zypp/repos.d/
collabnet-6.2.0.0.repo.cn_backup
```

20. If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
zypper ref
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) 👉 **Note:** If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

Insert the SuSE installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

If necessary, replace `sr0` with the identifier for your server's CD drive.

Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.

21. On the source code box, install the source code component of the TeamForge application.

```
zypper refresh
zypper install teamforge-scm
```

a) GIT: To install the GIT packages run the following command:

```
zypper install teamforge-git
```

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

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

a) Add the token *DEDICATED_INSTALL=true*

This makes the installation process very simple as theTeamForge installer takes care of configuring the apache, postgresql automatically without any manual intervention.

b) Use the *SCM_DEFAULT_SHARED_SECRET* on page 382 key from the `site-options.conf` file of the primary TeamForge server

c) Increase the value of MAX_WWW_CLIENTS.

```
MAX_WWW_CLIENTS=220
```

d) Configure the following settings if you're installing Git:

```
HOST_my.host.name=app etl database datamart subversion cvs
 indexer gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

Note: If an older version of TeamForge is integrated with Gerrit and if certain specific properties are customized directly in `gerrit.config>`, the customized properties are not migrated as they were configured when you upgrade to TeamForge 7.0

- Copy the value of teamforge.jumbopushThresold if it is not equal to "30" and add the following property to`site-options.conf` *GERRIT_GIT_PUSH_THRESHOLD* =<your_current_value_of "jumboPushThreshold"_from_gerrit.config>
- Copy the value of teamforge.refreshTimeout if it is not equal to "3600" and add the following property to `site-options.conf` property *GERRIT_GIT_REFRESH_PERIOD* =<your_current_value_of_"refreshTimeout"_from_gerrit.config>

23. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

24. 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.

25. Start TeamForge

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

26. Integrate Gerrit by running the `post-install.py` script.

```
/opt/collabnet/gerrit/scripts/post-install.py
```

a) Restart the Gerrit services.

```
/etc/init.d/collabnet restart gerrit
```

b) To verify the GIT integration:

Login to the app server and run the following command:

```
/etc/init.d/collabnet status
```

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Upgrade to TeamForge 7.0 on new hardware - SCM on a separate box

To upgrade to TeamForge 7.0, you install the new site and convert your existing site's data.

Important: There are no changes in the steps to upgrade to TeamForge 7.0 Patch 1

TeamForge 7.0 runs on Red Hat Enterprise Linux 6.3 and 6.4. To take advantage of the new operating system, you should install a fresh box and move your TeamForge 6.2 site to it.

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/9.0/backups/
teamforge_data_backup.dmp
exit
mkdir /tmp/backup_dir
cp /var/lib/pgsql/9.0/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/9.0/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 /sf-svnroot /opt/collabnet/teamforge/var /tmp/backup_dir
cd /tmp
tar czvf 62backup.tgz backup_dir
```

c) Back up your SSH keys, if any.

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

2. Stop TeamForge application server and Apache server.

```
/etc/init.d/apache2 stop
/etc/init.d/collabnet stop
```

3. Stop PostgreSQL.

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

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

```
scp /opt/collabnet/teamforge-installer/6.2.0.0/conf/site-options.conf
username@newbox:/tmp
```

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

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

```
scp /tmp/62backup.tgz username@newbox:/tmp
```

6. If the old server has the TeamForge CLI server add-on installed, then copy the cliserver.properties file under `/etc/` from the old server to`/shared/cliserver`

Note: Skip this step if you are migrating from TeamForge 6.2.0.1

```
sudo cp /etc/cliserver.properties /shared/teamforge/etc
```

Do this on the old TeamForge 6.2 source control box:

7. 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).
/gitroot	GIT source code repositories.

```
mkdir -p /tmp/scmbackup_dir/gerrit
cp -Rpf /svnroot /cvsroot /tmp/scmbackup_dir
```

If GIT integration is enabled:

```
mkdir /tmp/scmbackup_dir/gerrit
cp -Rpf /gitroot /tmp/scmbackup_dir
cp -Rpf /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
scmbackup_dir/gerrit
cd /tmp
tar czvf 62scmbackup.tgz scmbackup_dir
```

8. Stop TeamForge

```
/etc/init.d/apache2 stop
/etc/init.d/collabnet stop
```

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

```
scp /opt/collabnet/teamforge-installer/6.2.0.0/conf/site-options.conf
username@newbox:/tmp
```

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

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

```
scp /tmp/62scmbackup.tgz username@newscmbox:/tmp
```

Do this on the new TeamForge 7.0 app box

11. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
- See *the SuSE Linux Enterprise Server deployment guide* for help.

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

12. Check your basic networking setup.

See *Set up networking for your TeamForge box* on page 7 for details.

13. If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
zypper ref
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) **Note:** If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

Insert the SuSE installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

If necessary, replace `sr0` with the identifier for your server's CD drive.

Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.

14. Install the following application packages.

a) TeamForge: To install the TeamForge application packages run the following command:

```
zypper install teamforge
```

b) Black Duck Code Sight: To install Black Duck Code Sight run the following command:

```
zypper install teamforge-codesearch
```

15. Copy the `site-options.conf` file to the TeamForge installer directory.

```
cp /tmp/site-options.conf /opt/collabnet/teamforge-installer/7.0.0.1/conf
```

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

```
cd /tmp
tar xzvf 62backup.tgz
```

17. In the `site-options.conf` file, make sure you do the following:

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

a) Update the host name and domain name, if required.

HOST_my.host.name=app etl database datamart `indexer`
HOST_myscmbox.domain.name=subversion cvs

b) Check the*JAVA_HOME* variable to ensure that TeamForge is using the right JDK.

The JAVA_HOME variable should look like this:

```
JAVA_HOME=/usr/java/jdk1.7.0_21
```

c) If your site is running in SSL mode (SSL=on), add the following java runtime property to the *JBOSS_JAVA_OPTS* token.

```
JBOSS_JAVA_OPTS=-Xms1536m -Xmx1536m -XX:MaxPermSize=512m -server -XX:
+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/tmp -verbose:gc -XX:+
```

```
PrintGCTimeStamps -XX:+PrintGCDetails -Djsse.enableSNIExtension=false
 -Dsun.rmi.dgc.client.gcInterval=600000 -
Dsun.rmi.dgc.server.gcInterval=600000
```

d) **If the token *REQUIRE_PASSWORD_SECURITY* is enabled, then set a value for the token *PASSWORD_CONTROL_EFFECTIVE_DATE* to set the date from which the Password Control Kit (PCK) begins to disable, delete or expire user accounts. If the effective date is not set, the PCK starts disabling, deleting or expiring user accounts immediately.**

e) 👉 **Important:** If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

f) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

g) Add the token *DEDICATED_INSTALL=true* if it is not present in the `site-options.conf` file.

The dedicated installation mode is suitable when the server is completely dedicated to TeamForge application. This makes the installation process very simple as theTeamForge installer takes care of configuring your apache, postgresql automatically without any manual intervention. In a "Dedicated" install, it is advised not to install another software application other than TeamForge. It is also recommended not to run any unwanted services that are not used by TeamForge.

h) Update the host name and domain name, if required.

HOST_my.host.name=app database subversion cvs etl datamart `indexer`

i) Comment out the following tokens if they already enabled.

```
MAX_WWW_CLIENTS
```

```
PGSQL_MAX_CONNECTIONS
```

```
DATABASE_MAX_POOL_SIZE
```

```
SITE_DIR
```

j) Set the REPORTS_DATABASE_MAX_POOL_SIZE token as follows:

```
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

k) Increase the ETL heap size.

```
ETL_JAVA_OPTS=-Xms160m -Xmx512m -server -XX:+HeapDumpOnOutOfMemoryError
  -XX:HeapDumpPath=/tmp -verbose:gc -XX:+PrintGCTimeStamps -
XX:+PrintGCDetails -Dsun.rmi.dgc.client.gcInterval=600000 -
Dsun.rmi.dgc.server.gcInterval=600000
```

l) Set the USERS_WITH_NO_EXPIRY_PASSWORD token as follows:

```
USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer
```

m) Make sure that the following tokens have a value if ETL is enabled.

```
SOAP_ANONYMOUS_SHARED_SECRET=
ETL_SOAP_SHARED_SECRET=
```

n) Configure the following settings for Black Duck Code Sight.

```
HOST_mycodesearchbox.domain.com=codesearch
```

👉 **Note:** In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>

To enable SSL for Black Duck Code Sight, include this token:
BDCS_SSL=on
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
BDCS_ADMIN_PASSWORD=<password>
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

18. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

19. Start TeamForge.

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

👉 **Note:**

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

20. If the CLI reports are not configured with the earlier versions of TeamForge, run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

21. If the CLI reports are already configured with the earlier versions of TeamForge, follow these steps:

a) Backup the CLI reports.

```
sudo mkdir /root/backup
cd /root/backup
backup> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> sudo find . -name .svn | xargs sudo rm -rf
```

b) Remove the `cli/reports` folder from the branding repository.

👉 **Note:** Do a SVN delete to remove this folder from the repository.

```
sudo mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup>sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> sudo svn delete reports
cli> sudo svn commit -m "To delete the old CLI reports folder"
```

c) Run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

d) Commit the back-up 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```
sudo mkdir /root/restore
cd /root/restore
restore> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
sudo mkdir custom-reports
sudo cp -avx /root/backup/pkg custom-reports
sudo svn add custom-reports
sudo svn commit -m "adding the existing customized reports"
```

e) Manually schedule the cron job from the CLI command prompt.

```
/opt/collabnet/teamforge/add-ons/teamforge_cli/bin/ctf
ctf > server add <alise_name> <http/https://hostname>
ctf/list(2)> Now, hit the Enter key.
ctf > conn <alise_name>
```

You may be prompted to enter the username (<admin_username>) and password (<admin_password>)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```

22. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

23. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.0.

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

See *Synchronize TeamForge source control integrations* on page 251.

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 246.

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 239 for details.

f) To verify that the tracker initial load (triggered through `TrackerInitialJob`) completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

```
select * from etl_job where job_name ='tracker_initial_etl'
```

The status column should have a value of 1.

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

See *Customize anything on your site*.

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

See *Create a site-wide broadcast*.

Important: Do not delete the `teamforge-installer/7.0.0.1` directory. You will need it for future maintenance and upgrades.

Do this on the new TeamForge 7.0 SCM box

24. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
- See *the SuSE Linux Enterprise Server deployment guide* for help.

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

25. Check your basic networking setup.

See *Set up networking for your TeamForge box* on page 7 for details.

26. If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
zypper ref
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) **Note:** If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

Insert the SuSE installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

If necessary, replace `sr0` with the identifier for your server's CD drive.

Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.

27. On the source code box, install the source code component of the TeamForge application.

```
zypper ref
zypper install teamforge-scm
```

a) GIT: To install the GIT packages run the following command:

```
zypper install teamforge-git
```

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

```
vi /opt/collabnet/teamforge-installer/7.0.0.0/conf/site-options.conf
```

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

a) Add the token *DEDICATED_INSTALL=true*

This makes the installation process very simple as theTeamForge installer takes care of configuring the apache, postgresql automatically without any manual intervention.

b) Use the *SCM_DEFAULT_SHARED_SECRET* on page 382 key from the `site-options.conf` file of the primary TeamForge server

c) Increase the value of MAX_WWW_CLIENTS.

```
MAX_WWW_CLIENTS=220
```

d) Configure the following settings if you're installing Git:

```
HOST_my.host.name=app etl database datamart indexer gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

29. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

30. 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.

31. Convert your site data to work with TeamForge 7.0.

Tip: Before you kick off the data migration, use the `/etc/init.d/collabnet status` command to make sure the Jboss, Tomcat and Tomcatcs services 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 7.0 site.

32. Run the following post migration script.

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

33. Start TeamForge

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

34. Integrate Gerrit by running the `post-install.py` script.

```
/opt/collabnet/gerrit/scripts/post-install.py
```

a) Restart the Gerrit services.

```
/etc/init.d/collabnet restart gerrit
```

b) To verify the GIT integration:

Login to the app server and run the following command:

```
/etc/init.d/collabnet status
```

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Upgrade an advanced TeamForge site to TeamForge 7.0

Upgrading to TeamForge 7.0 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 "dedicate"d or "advanced"?* on page 223

Upgrade TeamForge to version 7.0 on the same box (Oracle)

To upgrade to TeamForge 7.0 without changing the box where your TeamForge where your pre-7.0 TeamForge site is running, you have to install the new TeamForge site and migrate the site data.

Important: There are no changes in the steps to upgrade to TeamForge 7.0 Patch 1

In this option, the Oracle database runs on one box, source control services runs on the second box and other services run on the application box.

Note: In a multi-box installation of TeamForge, all machines must have the same time zone - without this, ETL will not function properly.

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

1. Back up your site data.
   a) Make a dump file of your site database.

      To back up the Oracle database, follow the *Oracle backup procedure*.
   b) Make an archive file with the following data directories:

Directory	Contents
/opt/collabnet/teamforge/var	User-created data, such as artifact attachments
/sf-svnroot	Subversion repository for branding data

      ```
      mkdir /tmp/backup_dir
      cp -Rpf /sf-svnroot /opt/collabnet/teamforge/var /tmp/backup_dir
      cd /tmp
      tar czvf 62backup.tgz backup_dir
      ```
   c) Back up your SSH keys, if any.
   d) Back up your SSL certificates and keys, if any.
2. Stop the TeamForge application server and the Apache server, if they are running.

   ```
   /etc/init.d/apache2 stop
   /etc/init.d/collabnet stop
   ```
3. Move the repositories from any previous installs out of the way.

   ```
   mv /etc/zypp/repos.d/collabnet-6.2.0.0.repo /etc/zypp/repos.d/
   collabnet-6.2.0.0.repo.cn_backup
   ```
4. If you are installing TeamForge with internet access:
   - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
   - Copy it to `/etc/yum.repos.d/`.
   - Refresh your repository cache.

     ```
     zypper ref
     ```

   If you are installing TeamForge without internet access:
   a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) **Note:** If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

Insert the SuSE installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

If necessary, replace `sr0` with the identifier for your server's CD drive.

Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.

5. Install the TeamForge application.

```
zypper install teamforge-app teamforge-etl
```

6. In the `site-options.conf` file, make sure you do the following:

a) Update the host name and domain name, if necessary.

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

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

b) Check the*JAVA_HOME* variable to ensure that TeamForge is using the right JDK.

The JAVA_HOME variable should look like this:

```
JAVA_HOME=/usr/java/jdk1.7.0_21
```

c) If your site is running in SSL mode (SSL=on), add the following java runtime property to the *JBOSS_JAVA_OPTS* token.

```
JBOSS_JAVA_OPTS=-Xms1536m -Xmx1536m -XX:MaxPermSize=512m -server -XX:
+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/tmp -verbose:gc -XX:+
PrintGCTimeStamps -XX:+PrintGCDetails -Djsse.enableSNIExtension=false
 -Dsun.rmi.dgc.client.gcInterval=600000 -
Dsun.rmi.dgc.server.gcInterval=600000
```

d) **If the token *REQUIRE_PASSWORD_SECURITY* is enabled, then set a value for the token *PASSWORD_CONTROL_EFFECTIVE_DATE* to set the date from which the Password Control Kit (PCK) begins to disable, delete or expire user accounts. If the effective date is not set, the PCK starts disabling, deleting or expiring user accounts immediately.**

e) **Important:** If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

f) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

g) Comment out the following tokens if they already enabled.

```
MAX_WWW_CLIENTS
```

```
PGSQL_MAX_CONNECTIONS
```

```
DATABASE_MAX_POOL_SIZE
```

```
SITE_DIR
```

h) Set the REPORTS_DATABASE_MAX_POOL_SIZE token as follows:

```
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

i) Increase the ETL heap size.

```
ETL_JAVA_OPTS=-Xms160m -Xmx512m -server -XX:+HeapDumpOnOutOfMemoryError
  -XX:HeapDumpPath=/tmp -verbose:gc -XX:+PrintGCTimeStamps -
XX:+PrintGCDetails -Dsun.rmi.dgc.client.gcInterval=600000 -
Dsun.rmi.dgc.server.gcInterval=600000
```

j) Set the USERS_WITH_NO_EXPIRY_PASSWORD token as follows:

```
USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer
```

k) Configure the database and datamart read-only users settings.

- ```
  DATABASE_READ_ONLY_USER=sitedatabasereadonlyusername
  ```
- ```
  DATABASE_READ_ONLY_PASSWORD=sitedatabasereadonlyuserpwd
  ```
- ```
  REPORTS_DATABASE_READ_ONLY_USER=reportingreadonlyusername
  ```
- ```
  REPORTS_DATABASE_READ_ONLY_PASSWORD=reportingreadonlyuserpwd
  ```

l) Make sure that the following tokens have a value if ETL is enabled.

```
SOAP_ANONYMOUS_SHARED_SECRET=
ETL_SOAP_SHARED_SECRET=
```

m) Configure the following settings for Black Duck Code Sight.

```
HOST_mycodesearchbox.domain.com=codesearch
```

👉 **Note:** In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>

To enable SSL for Black Duck Code Sight, include this token:
BDCS_SSL=on
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
```

```
BDCS_ADMIN_PASSWORD=<password>
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

n) Make sure that the database service names of the production database and reports database are configured to *DATABASE_SERVICE_NAME* and *REPORTS_DATABASE_SERVICE_NAME* respectively.

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

7. **Note:** Execute this step only if 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 this command:

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

8. Run the following command to remove the pagespeed cache.

   ```
   cd /opt/collabnet/teamforge/cache
   rm -rf pagespeed
   ```

9. Log in as the Oracle user and create the read-only database user and permissions.

   a) Create the database read-only user that you will use to connect from CollabNet TeamForge.

   ```
   SQL> create user <database_readonly_user> identified by
   <database_readonly_password> default tablespace <your tablespace >
   temporary tablespace <temporary tablespace>;
   ```

   b) Grant the required permissions to the read-only user that you just created.

   ```
   SQL> grant create session to <database_readonly_user>;
   SQL> exit
   ```

   c) Create the datamart read-only user that you will use to connect from CollabNet TeamForge.

   ```
   SQL> create user <reports_readonly_user> identified by
   <reports_readonly_password> default tablespace <your tablespace>
   temporary tablespace <temporary tablespace>;
   ```

   d) Grant the required permissions to the datamart read-only user that you just created.

   ```
   SQL> grant create session to <reports_readonly_user>;
   SQL> exit
   ```

10. Recreate the runtime environment.

    ```
    cd /opt/collabnet/teamforge-installer/7.0.0.1
    ./install.sh -r -I -V
    ```

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 7.0.

    Tip: Before you kick off the data migration, use the `/etc/init.d/collabnet status` command to make sure the Jboss, Tomcat and Tomcatcs services 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 7.0 site.

13. Run the following post migration script.

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

14. Swap in the new Apache configuration file.

Note: Set the value of the Apache configuration parameters *MaxKeepAliveRequests=10000* and *MaxClients to 400* in the `/etc/apache2/httpd.conf` file before you restart Apache.

```
cd /etc/apache2
mv httpd.conf httpd.conf_old
cp httpd.conf.cn_new httpd.conf
cd /etc/sysconfig/
mv apache2 apache2_old
cp apache2.cn_new apache2
/etc/init.d/apache2 start
```

15. Start TeamForge.

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


Note: The sample TeamForge project template is installed as part of the collabnetstartup script. If the project template is installed, or it exists in the database by default, the startup script will not re-install it.

16. If the CLI reports are not configured with the earlier versions of TeamForge, run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

17. If the CLI reports are already configured with the earlier versions of TeamForge, follow these steps:

a) Backup the CLI reports.

```
sudo mkdir /root/backup
cd /root/backup
backup> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> sudo find . -name .svn | xargs sudo rm -rf
```

b) Remove the `cli/reports` folder from the branding repository.

Note: Do a SVN delete to remove this folder from the repository.

```
sudo mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup>sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> sudo svn delete reports
cli> sudo svn commit -m "To delete the old CLI reports folder"
```

c) Run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

d) Commit the back-up 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```
sudo mkdir /root/restore
cd /root/restore
restore> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
sudo mkdir custom-reports
sudo cp -avx /root/backup/pkg custom-reports
sudo svn add custom-reports
sudo svn commit -m "adding the existing customized reports"
```

e) Manually schedule the cron job from the CLI command prompt.

```
/opt/collabnet/teamforge/add-ons/teamforge_cli/bin/ctf
ctf > server add <alise_name> <http/https://hostname>
ctf/list(2)> Now, hit the Enter key.
ctf > conn <alise_name>
```

You may be prompted to enter the username (<admin_username>) and password (<admin_password>)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```

18. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

19. Integrate Black Duck Code Sight with TeamForge by running the Black Duck Code Sight `post-install.sh` script.

Note: It is assumed that Subversion's client configuration file (`/root/.subversion/config`) for the root user is the default one without customization.

Note: If the Black Duck Code Sight is running on a separate server, run the following command in the code sight server.

```
sudo /opt/collabnet/teamforge/runtime/scripts/codesearch/post-install.sh
```

20. After you run the Black Duck Code Sight `post-install.sh` script, run the following script from the same directory:

```
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL similar to "http://myint.box.net/svn/repos" where `myint.box` is the box with the SCM integration server.

Also, if you add a new integration server at some point later, you'll need to run this script, (after creating new integration server), in the TeamForge `app` box.

Important: Do this if you have a multi-box setup with a SCM integration server on a separate server.

21. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.0.

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

See *Synchronize TeamForge source control integrations* on page 251.

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 246.

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 239 for details.

f) To verify that the tracker initial load (triggered through `TrackerInitialJob`) completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

```
select * from etl_job where job_name ='tracker_initial_etl'
```

The status column should have a value of 1.

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

See *Customize anything on your site*.

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

See *Create a site-wide broadcast*.

Important: Do not delete the `teamforge-installer/7.0.0.1` directory. You will need it for future maintenance and upgrades.

Do the following to upgrade the SCM box (CVS/Subversion/Git)

22. 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.

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

Directory	Contents
/svnroot	SVN source code repositories.
/cvsroot	CVS source code repositories (not present on all sites).
opt/collabnet/gerrit/etc	opt/collabnet/gerrit/etc.
/opt/collabnet/gerrit/.ssh	Gerrit SSH Keys.

```
mkdir -p /tmp/scmbackup_dir
cp -Rpf /svnroot /cvsroot /tmp/scmbackup_dir
mkdir -p /tmp/scmbackup_dir/gerrit
cp -Rpf /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
scmbackup_dir/gerrit
tar czvf 62scmbackup.tgz scmbackup_dir
```

b) Back up your SSH keys, if any.

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

23. Stop TeamForge

```
/etc/init.d/apache2 stop
/etc/init.d/collabnet stop tomcat
```

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

```
mv /etc/zypp/repos.d/collabnet-6.2.0.0.repo /etc/zypp/repos.d/
collabnet-6.2.0.0.repo.cn_backup
```

25. If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
zypper ref
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) **Note:** If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

Insert the SuSE installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

If necessary, replace `sr0` with the identifier for your server's CD drive.

Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.

26. On the source code box, install the source code component of the TeamForge application.

```
zypper refresh
zypper install teamforge-scm
```

a) GIT: To install the GIT packages run the following command:

```
zypper install teamforge-git
```

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

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

a) Use the *SCM_DEFAULT_SHARED_SECRET* on page 382 key from the `site-options.conf` file of the primary TeamForge server

b) Increase the value of MAX_WWW_CLIENTS.

```
MAX_WWW_CLIENTS=220
```

c) Configure the following settings if you're installing Git:

```
HOST_my.host.name=app etl database datamart subversion cvs
 indexer gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

Note: If an older version of TeamForge is integrated with Gerrit and if certain specific properties are customized directly in `gerrit.config>`, the customized properties are not migrated as they were configured when you upgrade to TeamForge 7.0

- Copy the value of teamforge.jumbopushThresold if it is not equal to "30" and add the following property to`site-options.conf` *GERRIT_GIT_PUSH_THRESHOLD* =<your_current_value_of "jumboPushThreshold"_from_gerrit.config>
- Copy the value of teamforge.refreshTimeout if it is not equal to "3600" and add the following property to `site-options.conf` property *GERRIT_GIT_REFRESH_PERIOD* =<your_current_value_of_"refreshTimeout"_from_gerrit.config>

28. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

29. 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.

30. Start TeamForge

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

31. Integrate Gerrit by running the `post-install.py` script.

```
/opt/collabnet/gerrit/scripts/post-install.py
```

a) Restart the Gerrit services.

```
/etc/init.d/collabnet restart gerrit
```

b) To verify the GIT integration:

Login to the app server and run the following command:

```
/etc/init.d/collabnet status
```

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Upgrade TeamForge to version 7.0 on new hardware (Oracle)

To upgrade to TeamForge 7.0, set up a new box, then bring over your old site's data and convert it.

Important: There are no changes in the steps to upgrade to TeamForge 7.0 Patch 1

In this option, the Oracle database runs on one box, source control services run on the second box and other services run on the application box. TeamForge 7.0 runs on SuSE Linux Enterprise Server 11 SP2. To take advantage of the new operating system, you should install a fresh box and move your TeamForge 6.1.1 site to it.

Note: In a multi-box installation of TeamForge, all machines must have the same time zone - without this, ETL will not function properly.

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

1. Back up your site data.

a) Make a dump file of your site database.

To back up the Oracle database, follow the *Oracle backup procedure*.

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

Directory	Contents
/opt/collabnet/teamforge/var	User-created data, such as artifact attachments
/sf-svnroot	Subversion repository for branding data

```
mkdir /tmp/backup_dir
cp -Rpf /sf-svnroot /opt/collabnet/teamforge/var /tmp/backup_dir
cd /tmp
tar czvf 62backup.tgz backup_dir
```

c) Back up your SSH keys, if any.

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

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

```
/etc/init.d/apache2 stop
/etc/init.d/collabnet stop
```

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

```
scp /tmp/62backup.tgz username@newbox:/tmp
```

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

```
scp /opt/collabnet/teamforge-installer/6.2.0.0/conf/site-options.conf
username@newbox:/tmp
```

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

Do this on the TeamForge 6.2 source control box

5. 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)
/opt/collabnet/gerrit/etc	Geritt Data
/opt/collabnet/gerrit/.ssh	Gerrit SSH Keys

```
mkdir -p /tmp/scmbackup_dir/gerrit
cp -Rpf /svnroot /cvsroot /tmp/scmbackup_dir
cp -Rpf /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
scmbackup_dir/gerrit
tar czvf 62scmbackup.tgz scmbackup_dir
```

   b) Back up your SSH keys, if any.
   c) Back up your SSL certificates and keys, if any.
6. Stop TeamForge.

```
/etc/init.d/apache2 stop
/etc/init.d/collabnet stop tomcat
```

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

```
sudo scp /opt/collabnet/teamforge-installer/6.2.0.0/conf/site-options.conf
username@newscmbox:/tmp
```

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

```
scp /tmp/scm62backup.tgz username@newscmbox:/tmp
```

Do these steps on the new TeamForge 7.0 application box:

9. If you are installing TeamForge with internet access:

   - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
   - Copy it to `/etc/yum.repos.d/`.
   - Refresh your repository cache.

```
zypper ref
```

   If you are installing TeamForge without internet access:

   a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) **Note:** If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

Insert the SuSE installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

If necessary, replace `sr0` with the identifier for your server's CD drive.

Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.

10. Download the TeamForge 7.0 installation repository from *open.collab.net*. Copy it to `/etc/zypp/repos.d/`.

11. Install the TeamForge application.

```
zypper install teamforge
```

12. Copy the `site-options.conf` file to the TeamForge installer directory.

```
cp /tmp/site-options.conf /opt/collabnet/teamforge-installer/7.0.0.1/conf
```

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

```
cd /tmp
tar xzvf 62backup.tgz
```

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

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

15. In the `site-options.conf` file, make sure you do the following:

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

a) Check the*JAVA_HOME* variable to ensure that TeamForge is using the right JDK.

The JAVA_HOME variable should look like this:

```
JAVA_HOME=/usr/java/jdk1.7.0_21
```

b) If your site is running in SSL mode (SSL=on), add the following java runtime property to the *JBOSS_JAVA_OPTS* token.

```
JBOSS_JAVA_OPTS=-Xms1536m -Xmx1536m -XX:MaxPermSize=512m -server -XX:
+HeapDumpOnOutOfMemoryError -XX:HeapDumpPath=/tmp -verbose:gc -XX:+
PrintGCTimeStamps -XX:+PrintGCDetails -Djsse.enableSNIExtension=false
 -Dsun.rmi.dgc.client.gcInterval=600000 -
Dsun.rmi.dgc.server.gcInterval=600000
```

c) **If the token *REQUIRE_PASSWORD_SECURITY* is enabled, then set a value for the token *PASSWORD_CONTROL_EFFECTIVE_DATE* to set the date from which the Password Control Kit (PCK) begins to disable, delete or expire user accounts. If the effective date is not set, the PCK starts disabling, deleting or expiring user accounts immediately.**

d) 👉 **Important:** If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

e) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

f) Comment out the following tokens if they already enabled.

```
MAX_WWW_CLIENTS
```

```
PGSQL_MAX_CONNECTIONS
```

```
DATABASE_MAX_POOL_SIZE
```

```
SITE_DIR
```

g) Set the REPORTS_DATABASE_MAX_POOL_SIZE token as follows:

```
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

h) Increase the ETL heap size.

```
ETL_JAVA_OPTS=-Xms160m -Xmx512m -server -XX:+HeapDumpOnOutOfMemoryError
  -XX:HeapDumpPath=/tmp -verbose:gc -XX:+PrintGCTimeStamps -
XX:+PrintGCDetails -Dsun.rmi.dgc.client.gcInterval=600000 -
Dsun.rmi.dgc.server.gcInterval=600000
```

i) Set the USERS_WITH_NO_EXPIRY_PASSWORD token as follows:

```
USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer
```

j) Configure the database and datamart read-only users settings.

- `DATABASE_READ_ONLY_USER=sitedatabasereadonlyusername`
- `DATABASE_READ_ONLY_PASSWORD=sitedatabasereadonlyuserpwd`
- `REPORTS_DATABASE_READ_ONLY_USER=reportingreadonlyusername`
- `REPORTS_DATABASE_READ_ONLY_PASSWORD=reportingreadonlyuserpwd`

k) Make sure that the following tokens have a value if ETL is enabled.

```
SOAP_ANONYMOUS_SHARED_SECRET=
ETL_SOAP_SHARED_SECRET=
```

l) Configure the following settings for Black Duck Code Sight.

```
HOST_mycodesearchbox.domain.com=codesearch
```

👉 **Note:** In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>
```

```
To enable SSL for Black Duck Code Sight, include this token:
BDCS_SSL=on
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
BDCS_ADMIN_PASSWORD=<password>
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

m) Make sure that the database service names of the production database and reports database are configured to *DATABASE_SERVICE_NAME* and *REPORTS_DATABASE_SERVICE_NAME* respectively.

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

16. **Note:** Execute this step only if 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 this command:

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

17. Log in as the Oracle user and create the read-only database user and permissions.

a) Create the database read-only user that you will use to connect from CollabNet TeamForge.

```
SQL> create user <database_readonly_user> identified by
<database_readonly_password> default tablespace <your tablespace >
temporary tablespace <temporary tablespace>;
```

b) Grant the required permissions to the read-only user that you just created.

```
SQL> grant create session to <database_readonly_user>;
SQL> exit
```

c) Create the datamart read-only user that you will use to connect from CollabNet TeamForge.

```
SQL> create user <reports_readonly_user> identified by
<reports_readonly_password> default tablespace <your tablespace>
temporary tablespace <temporary tablespace>;
```

d) Grant the required permissions to the datamart read-only user that you just created.

```
SQL> grant create session to <reports_readonly_user>;
SQL> exit
```

18. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

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 7.0.

Tip: Before you kick off the data migration, use the `/etc/init.d/collabnet status` command to make sure the Jboss, Tomcat and Tomcatcs services 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 7.0 site.

21. Run the following post migration script.

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

22. Swap in the new Apache configuration file.

Note: Set the value of the Apache configuration parameters *MaxKeepAliveRequests=10000* and *MaxClients to 400* in the `/etc/apache2/httpd.conf` file before you restart Apache.

```
cd /etc/apache2
mv httpd.conf httpd.conf_old
cp httpd.conf.cn_new httpd.conf
cd /etc/sysconfig/
mv apache2 apache2_old
cp apache2.cn_new apache2
/etc/init.d/apache2 start
```

23. Start TeamForge.

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

Note: The sample TeamForge project template is installed as part of the collabnetstartup script. If the project template is installed, or it exists in the database by default, the startup script will not re-install it.

24. If the CLI reports are not configured with the earlier versions of TeamForge, run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

25. If the CLI reports are already configured with the earlier versions of TeamForge, follow these steps:

a) Backup the CLI reports.

```
sudo mkdir /root/backup
cd /root/backup
backup> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> sudo find . -name .svn | xargs sudo rm -rf
```

b) Remove the `cli/reports` folder from the branding repository.

Note: Do a SVN delete to remove this folder from the repository.

```
sudo mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup>sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> sudo svn delete reports
cli> sudo svn commit -m "To delete the old CLI reports folder"
```

c) Run the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

d) Commit the back-up 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```
sudo mkdir /root/restore
cd /root/restore
restore> sudo svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
sudo mkdir custom-reports
```

```
sudo cp -avx /root/backup/pkg custom-reports
sudo svn add custom-reports
sudo svn commit -m "adding the existing customized reports"
```

e) Manually schedule the cron job from the CLI command prompt.

```
/opt/collabnet/teamforge/add-ons/teamforge_cli/bin/ctf
ctf > server add <alise_name> <http/https://hostname>
ctf/list(2)> Now, hit the Enter key.
ctf > conn <alise_name>
```

You may be prompted to enter the username (<admin_username>) and password (<admin_password>)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```

26. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

27. Integrate Black Duck Code Sight with TeamForge by running the Black Duck Code Sight `post-install.sh` script.

Note: It is assumed that Subversion's client configuration file (`/root/.subversion/config`) for the root user is the default one without customization.

Note: If the Black Duck Code Sight is running on a separate server, run the following command in the code sight server.

```
sudo /opt/collabnet/teamforge/runtime/scripts/codesearch/post-install.sh
```

28. After you run the Black Duck Code Sight `post-install.sh` script, run the following script from the same directory:

```
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL similar to "http://myint.box.net/svn/repos" where `myint.box` is the box with the SCM integration server.

Also, if you add a new integration server at some point later, you'll need to run this script, (after creating new integration server), in the TeamForge `app` box.

Important: Do this if you have a multi-box setup with a SCM integration server on a separate server.

29. If the token *VALIDATE_SSL_CERTS* is set to "true", you must run the codesearch runtime script *trust-cert.sh* in the application box.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
```

30. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.0.

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

See *Synchronize TeamForge source control integrations* on page 251.

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 246.

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 239 for details.

f) To verify that the tracker initial load (triggered through `TrackerInitialJob`) completed successfully, run the following SQL from `[RUNTIME_DIR]/scripts/psql-reporting-wrapper` or by by selecting the **Datamart** option on the **System Tools** > **Ad Hoc Database Query** page in the TeamForge web interface.

```
select * from etl_job where job_name ='tracker_initial_etl'
```

The status column should have a value of 1.

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

See *Customize anything on your site*.

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

See *Create a site-wide broadcast*.

Important: Do not delete the `teamforge-installer/7.0.0.1` directory. You will need it for future maintenance and upgrades.

Do this on the new TeamForge 7.0 source control box

31. Check your basic networking setup.

See *Set up networking for your TeamForge box* on page 7 for details.

32. Stop TeamForge

```
/etc/init.d/apache2 stop
/etc/init.d/collabnet stop tomcat
```

33. If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
zypper ref
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) **Note:** If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

Insert the SuSE installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

If necessary, replace `sr0` with the identifier for your server's CD drive.

Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.

34. On the source code box, install the source code component of the TeamForge application.

```
zypper refresh
zypper install teamforge-scm
```

a) GIT: To install the GIT packages run the following command:

```
zypper install teamforge-git
```

35. Copy the `site-options.conf` file to the TeamForge installer directory.

```
cp /tmp/site-options.conf /opt/collabnet/teamforge-installer/7.0.0.1/conf
```

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

```
cd /tmp
tar xzvf 62scmbackup.tgz
```

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

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

a) Use the *SCM_DEFAULT_SHARED_SECRET* on page 382 key from the `site-options.conf` file of the primary TeamForge server

b) Increase the value of MAX_WWW_CLIENTS.

```
MAX_WWW_CLIENTS=220
```

c) Configure the following settings if you're installing Git:

```
HOST_my.host.name=app etl database datamart subversion cvs
 indexer gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

Note: If an older version of TeamForge is integrated with Gerrit and if certain specific properties are customized directly in `gerrit.config>`, the customized properties are not migrated as they were configured when you upgrade to TeamForge 7.0

- Copy the value of teamforge.jumbopushThresold if it is not equal to "30" and add the following property to`site-options.conf` *GERRIT_GIT_PUSH_THRESHOLD* =<your_current_value_of "jumboPushThreshold"_from_gerrit.config>
- Copy the value of teamforge.refreshTimeout if it is not equal to "3600" and add the following property to `site-options.conf` property *GERRIT_GIT_REFRESH_PERIOD* =<your_current_value_of_"refreshTimeout"_from_gerrit.config>

38. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

39. Bring your site data back.

```
Reload svnroot,cvsroot,gitroot (if GIT is enabled)
```

40. 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.

41. Start TeamForge

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

42. Integrate Gerrit by running the `post-install.py` script.

```
/opt/collabnet/gerrit/scripts/post-install.py
```

a) Restart the Gerrit services.

```
/etc/init.d/collabnet restart gerrit
```

b) To verify the GIT integration:

Login to the app server and run the following command:

```
/etc/init.d/collabnet status
```

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Upgrade to CollabNet TeamForge 7.0 on a virtual machine

You can upgrade to CollabNet TeamForge 7.0 from a VMware installation of CollabNet TeamForge 6.2.

Note:

- To upgrade from any version earlier than 6.2 (including CollabNet SourceForge Enterprise 5.x or any version of SourceForge Enterprise Edition), first *upgrade your site to TeamForge 6.2* and start it up. Then follow these steps for upgrading to CollabNet TeamForge 7.0.
- When you upgrade to TeamForge 7.0, the VMware updater automatically upgrades your site to the latest patch release, if available.

For basic installation and configuration on VMware, see these instructions:

- *Get TeamForge 7.0 for VMware Player or VMware ESXi*
- *Configure CollabNet TeamForge on VMware Player*

1. On the computer where your previous version of TeamForge is running, open a command prompt and log onto the machine as an administrator.

   Note: Type `N` when prompted to start the TeamForge configuration tool.

2. Get the VMware updater for TeamForge 7.0 from your CollabNet Support representative and unzip it in the `/tmp` directory.

```
unzip updater-teamforge-vmware-7.0.0.0.zip
```

3. Back up your TeamForge 6.2 site.

   See *Back up CollabNet TeamForge data* on page 237 for instructions.

4. Review the key configuration variables for your TeamForge site.

   The configuration file is at `/opt/collabnet/teamforge/runtime/conf/runtime-options.conf`. Note these key variables and their values:

```
HOST_<host_name>
DOMAIN_<host_name>
SYSTEM_EMAIL
ADMIN_EMAIL
JAMES_POSTMASTER_EMAIL
JAMES_GATEWAY_HOST
JAMES_GATEWAY_PORT
```

5. Run the upgrade utility.

```
cd updater-teamforge-vmware-7.0.0.0
./updater-6_2-7_0.sh
```

The upgrade can take several minutes to complete, depending on system resources.

Important: Type `Y` when prompted.

6. **Note:** Skip this step if you are not integrating TeamForge with Gerrit.

   If you want to integrate Gerrit, run the `post-install.py` script. Keep the TeamForge site administrator's user name and password handy before running the `post-install.py` script.

   ```
   /opt/collabnet/gerrit/scripts/post-install.py
   ```

   a) When prompted, type the user name and password of the dedicated TeamForge site administrator's user account (non-expiring and unlockable administrator account).
   b) Type a new Gerrit database password. You may note down the Gerrit database password for future use.
   c) Restart the Gerrit services.

   ```
   /etc/init.d/collabnet restart gerrit
   ```

7. Configure CLI reports.

   ```
   /opt/collabnet/teamforge/runtime/scripts/post-install.py
   ```

Configure the CLI Jobs Linked Application's URL

Important: Configure the CLI Jobs linked application's URL only if the site option token *HOST_localhost* is configured as "localhost" in the `site-options.conf` file.

8. After logging on to TeamForge, click the **Projects** tab.
9. Click the **Look** project.
10. Click **Project Admin** in the project navigation bar.
11. Click **Project Toolbar** from the **Project Admin Menu**.
12. Select the **Linked Applications** tab.
13. Select the **CLI Jobs** linked application's check box and click **Edit**.
14. Modify the URL: Replace the "`localhost`" with the hostname or IP address of the computer that hosts the TeamForge application.
15. Click **Save**.

Troubleshooting: VMware installer failed to install new TeamForge version

When your TeamForge 7.0 upgrade process on VMware fails during the installation phase, something may be blocking the installer or the data migration utility from running TeamForge from your VMware Player. Try these manual steps to recover.

Execute these steps from a command line terminal inside the VMware Player.

1. Ensure that the collabnet-7.0.0.1.repo file is available in the `/etc/yum.repos.d`
2. Install the TeamForge application packages.

   ```
   yum install teamforge
   ```

3. Create the runtime environment.

   ```
   cd /opt/collabnet/teamforge-installer/7.0.0.1
   ./install.sh -r -I -n -V -d /opt/collabnet/teamforge -F
   ```

4. Run the data migration utility.

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

5. Start TeamForge.

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

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.

- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

Troubleshooting: VMware installer failed to finish TeamForge 7.0 installation

When your TeamForge 7.0 upgrade process on VMware fails during the post-install phase, it may have been unable to configure the installation properly. Try these manual steps to recover.

Execute these steps from a command line terminal inside the VMware player.

1. Set the permissions for the branding repository.

```
chown -R apache:apache /opt/collabnet/teamforge/var/overrides
chown -R apache:apache /sf-svnroot
```

2. Replace the automatic configuration script.

```
cd $UPDATE_DIR (UPDATE_DIR=location where vmware updater is extracted)
cp -f configure-teamforge.py /root/configure-teamforge.py
```

3. Replace the static HTML content.

```
cd $UPDATE_DIR
'cp' $UPDATE_DIR/static-content/index.html /var/www/html
'cp' $UPDATE_DIR/static-content/logo.gif /var/www/html/static_files/
'cp' $UPDATE_DIR/static-content/static.htm /var/www/html
'cp' $UPDATE_DIR/static-content/c.css /var/www/html/static_files/
'cp' $UPDATE_DIR/static-content/initial_tasks.png /var/www/html/_i/
'cp' $UPDATE_DIR/static-content/launch.gif /var/www/html/_i
```

Is my TeamForge site "dedicated" or "advanced?"

Check the value of the *DEDICATED_INSTALL* variable to see if you are working with a dedicated or an advanced site.

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/7.0.0.1/conf/site-options.conf
```

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 "dedicate"d or "advanced"?* on page 223

If you do have a dedicated site, this may help lighten your load a little.

Enable reporting while upgrading from TeamForge 6.1.1 to 7.0

When upgrading from a TeamForge 6.1.1 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 7.0

The following procedure is an outline of what you need to when upgrading from TeamForge 6.1.1 (no reporting) to TeamForge 7.0 (with reporting).

1. Stop TeamForge.
2. Back up your site data.
3. Install the TeamForge 7.0 build, making sure that you change the 6.1 site-options for the `jdk` and `MaxPermSize` requirements in 7.0.

   Note: Tokens related to reporting should not be enabled.
4. Migrate the data to TeamForge 7.0.
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. You need to add the SSL certificate to the Java keystore if SSL is set to on, the site uses a self-signed certificate and VALIDATE_SSL_CERTS is set to "true".

   For instructions on adding the self-signed certificate to the Java keystore, see *Protect integrations with SSL* on page 248.
9. Start the TeamForge 7.0 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 TeamForge 7.0

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 7.0* on page 118.

1. Stop TeamForge.

   ```
   /etc/init.d/httpd stop
   /etc/init.d/apache2 stop
   /etc/init.d/postgresql-9.0 stop /etc/init.d/postgresql stop
   /etc/init.d/collabnet stop
   ```

2. Install the TeamForge application.

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

   ```
   zypper install teamforge
   ```

   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/7.0.0.1
   ```

   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
   ```

   Note:

   - The TeamForge startup script installs the sample TeamForge project templates by default.
   - If the project templates are already installed, you cannot re-install them using the TeamForge startup script.

- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

TeamForge 7.0 Patch

TeamForge 7.0 Patch 1 is available. Apply this patch if you have TeamForge 7.0 up and running.

Note:

TeamForge add-on packages can remain installed when a patch for TeamForge 7.0 is applied. No updates are required. However, you should verify the compatibility of Add-ons applied to your site as some may be more specific to your site.

The most common Add-ons and their TeamForge compatibility are published here: *https://ctf.open.collab.net/sf/go/page1363*

If you have any questions regarding compatibility or updating a specific add-on, contact *CollabNet Support* for more details.

Patch CollabNet TeamForge 7.0 on Red Hat/CentOS

Use this information to install or downgrade a TeamForge patch on RedHat/CentOS.

Apply a patch on RHEL / CentOS

Use these instructions to patch your TeamForge 7.0 site with TeamForge 7.0 Patch 1.

Important: You must log in as a root user to apply this patch.

1. Stop TeamForge

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

   Note: In case of a multi-box setup, run the above commands in the respective boxes.

2. Install the TeamForge application patch.

```
yum install teamforge
```

   The yum command installs all the TeamForge patches available in the Collabnet yum repository for TeamForge 7.0.

   Note: When you upgrade, the `site-options.conf` file is automatically backed up into `/opt/collabnet/teamforge-installer/7.0.0.1/conf` directory.

   You can upgrade TeamForge based on the services enabled in the respective boxes in case of an advanced mode setup. For example, to install the TeamForge patches run the command `yum install teamforge-scm` in case of "SCM on a separate box".

3. If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.
4. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

5. Run the following post migration script.

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

6. If you have Git integration, run the following command.

```
ln -sf /opt/collabnet/gerrit/bin/gerrit.sh /etc/init.d/gerrit
```

7. If your TeamForge site is running in advanced mode, start the external services by running the following commands.
   - Start the Apache server.

     ```
     /etc/init.d/httpd start
     ```
   - If the TeamForge instance uses PostgreSQL database, start the PostgreSQL.

     ```
     /etc/init.d/postgresql-9.0 start
     ```
8. Start TeamForge.

   ```
   /etc/init.d/collabnet start
   ```
9. Run the following `post-install.py` script.

   ```
   /opt/collabnet/teamforge/runtime/scripts/post-install.py
   ```

To verify if the patch is working:

Click **Admin** in the site navigation bar.
On the site administration navigation bar, click **System Tools**.
Click **Build Information** and observe the following:

Version: 7.0.0.1
Applied Patches: 380.1: CN-collabnet-core,CN-migration,CN-runtime

Downgrade a patch on RHEL / CentOS

To downgrade a patch that you have installed on your TeamForge site, run the patch installer from the directory corresponding to the patch.

Important: You must log in as a root user to apply this patch.

1. Stop TeamForge.

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

   Note: In case of multi-box setup, run the above commands in the respective boxes.
2. Run the downgrade command.

   Note:
   - When you downgrade, the backup of the `site-options.conf` file will be available in `/opt/collabnet/teamforge-installer/7.0.0.0/conf` directory.
   - You must downgrade TeamForge based on the services enabled in the box in case of an advanced mode setup. For example, run the downgrade command `yum downgrade teamforge-scm CN-collabnet-core CN-migration CN-runtime TeamForge-installer` in case of "SCM on a separate box".

   ```
   yum downgrade CN-collabnet-core CN-runtime CN-migration teamforge
   TeamForge-installer teamforge-app teamforge-database teamforge-etl
   teamforge-scm teamforge-codesearch ctf_clireports
   ```

   If you have git integration, add `"teamforge-git"` to the `yum downgrade` command as follows:

   ```
   yum downgrade CN-collabnet-core CN-runtime CN-migration teamforge
   TeamForge-installer teamforge-app teamforge-database teamforge-etl
   teamforge-scm teamforge-codesearch teamforge-git ctf_clireports
   ```
3. Recreate the run time environment.

   ```
   cd /opt/collabnet/teamforge-installer/7.0.0.0
   ```

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

4. If you have Git integration, run the following command.

```
ln -sf /opt/collabnet/gerrit/bin/gerrit.sh /etc/init.d/gerrit
```

5. If your TeamForge site is running in advanced mode, start the external services by running the following commands.
   - Start the Apache server.

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

   - If the TeamForge instance uses PostgreSQL database, start the PostgreSQL.

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

6. Start TeamForge.

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

To verify if the patch has been reverted:

Click **Admin** in the site navigation bar.
On the site administration navigation bar, click **System Tools**.
Click **Build Information** and observe the following:

Version: 7.0.0.0
Applied Patches: None

Patch CollabNet TeamForge 7.0 on SuSE

Use this information to install or downgrade a TeamForge patch on SuSE.

Apply a patch on SuSE

Use these instructions to patch your TeamForge 7.0 site with TeamForge 7.0 Patch 1.

Important: You must log in as a root user to apply this patch.

1. Stop TeamForge.

```
/etc/init.d/apache2 stop
/etc/init.d/postgresql stop
/etc/init.d/collabnet stop
```

Note: In case of a multi-box setup, run the above commands in the respective boxes.

2. Install the patch.

```
zypper install teamforge
```

The zypper command, installs all the TeamForge patches available in the Collabnet yum repository for TeamForge 7.0.

Note: When you upgrade, the`site-options.conf` file is automatically backed up in to the `/opt/collabnet/teamforge-installer/7.0.0.1/conf` directory.

You can upgrade TeamForge based on the services enabled in the respective boxes in case of an advanced mode setup. For example, to install the TeamForge patches, run the command `zypper install teamforge-scm` in case of "SCM on a separate box".

3. If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.
4. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

5. Run the following post migration script.

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

6. If you have Git integration, run the following command.

   ```
   ln -sf /opt/collabnet/gerrit/bin/gerrit.sh /etc/init.d/gerrit
   ```

7. If your TeamForge site is running in advanced mode, start the external services by running the following commands.

   - Start the Apache server.

     ```
     /etc/init.d/apache2 start
     ```

   - If the TeamForge instance uses PostgreSQL database, start the PostgreSQL.

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

8. Start TeamForge.

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

9. Run the following `post-install.py` script.

   ```
   /opt/collabnet/teamforge/runtime/scripts/post-install.py
   ```

To verify if the patch is working:

Click **Admin** in the site navigation bar.
On the site administration navigation bar, click **System Tools**.
Click **Build Information** and observe the following:

Version: 7.0.0.1
Applied Patches: 380.1: CN-collabnet-core,CN-migration,CN-runtime

Downgrade a patch on SuSE

To downgrade a patch that you have installed on your TeamForge site, run the patch installer from the directory corresponding to the patch.

Important: You must log in as a root user to apply this patch.

1. Stop TeamForge.

   ```
   /etc/init.d/apache2 stop
   /etc/init.d/postgresql stop
   /etc/init.d/collabnet stop
   ```

   Note: In case of multi-box setup, run the above commands in the respective boxes.

2. Downgrade the patch.

   ```
   zypper install --force teamforge-7.0.0.0-380
   ```

   During installation, you will be provided with three options at the prompt. Select the "downgrade" option to downgrade the "TeamForge and CN packages.

   Note: During upgrade, the `site-options.conf` file is automatically backed up into `/opt/collabnet/teamforge-installer/7.0.0.0/`

   You can downgrade TeamForge based on the services enabled in the box in case of an advanced mode setup. For example, run the command `zypper install --force teamforge-scm-7.0.0.0-380` in case of "SCM on a separate box".

3. Recreate the runtime environment.

   ```
   cd /opt/collabnet/teamforge-installer/7.0.0.0
   ./install.sh -r -V -I
   ```

4. If you have Git integration, run the following command.

```
ln -sf /opt/collabnet/gerrit/bin/gerrit.sh /etc/init.d/gerrit
```

5. If your TeamForge site is running in advanced mode, start the external services by running the following commands.
   - Start the Apache server.

     ```
     /etc/init.d/apache2 start
     ```
   - If the TeamForge instance uses PostgreSQL database, start the PostgreSQL.

     ```
     /etc/init.d/postgresql start
     ```
6. Start TeamForge.

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

To verify if the patch has been reverted:

Click **Admin** in the site navigation bar.
On the site administration navigation bar, click **System Tools**.
Click **Build Information** and observe the following:

Version: 7.0.0.0
Applied Patches: None

Patch TeamForge 7.0 on VMware Player or VMware ESXi

Use this information to apply or downgrade a TeamForge patch on VMware Player or VMware ESXi .

Apply a patch on VMware Player or VMware ESXi

Use these instructions to patch your TeamForge 7.0 site with TeamForge 7.0 Patch 1.

Important: You must log in as a root user to apply this patch.

1. Stop TeamForge

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

   Note: In case of a multi-box setup, run the above commands in the respective boxes.
2. Install the TeamForge application patch.

   ```
   yum install teamforge
   ```

   The yum command installs all the TeamForge patches available in the Collabnet yum repository for TeamForge 7.0.

   Note: When you upgrade, the `site-options.conf` file is automatically backed up into `/opt/collabnet/teamforge-installer/7.0.0.1/conf` directory.

   You can upgrade TeamForge based on the services enabled in the respective boxes in case of an advanced mode setup. For example, to install the TeamForge patches run the command `yum install teamforge-scm` in case of "SCM on a separate box".
3. Recreate the runtime environment.

   ```
   cd /opt/collabnet/teamforge-installer/7.0.0.1
   ./install.sh -r -I -V
   ```
4. Run the following post migration script.

   ```
   /opt/collabnet/teamforge/runtime/scripts/post-migrate.py
   ```
5. If you have Git integration, run the following command.

```
ln -sf /opt/collabnet/gerrit/bin/gerrit.sh /etc/init.d/gerrit
```

6. If your TeamForge site is running in advanced mode, start the external services by running the following commands.
   - Start the Apache server.

     ```
     /etc/init.d/httpd start
     ```
   - If the TeamForge instance uses PostgreSQL database, start the PostgreSQL.

     ```
     /etc/init.d/postgresql-9.0 start
     ```
7. Start TeamForge.

   ```
   /etc/init.d/collabnet start
   ```
8. Run the following `post-install.py` script.

   ```
   sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
   ```

To verify if the patch is working:

Click **Admin** in the site navigation bar.
On the site administration navigation bar, click **System Tools**.
Click **Build Information** and observe the following:

Version: 7.0.0.1
Applied Patches: 380.1: CN-collabnet-core,CN-migration,CN-runtime

Downgrade a patch on VMware Player or VMware ESXi

To downgrade a patch that you have installed on your TeamForge site, run the patch installer from the directory corresponding to that patch.

Important: You must log in as a root user to apply this patch.

1. Stop TeamForge.

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

   Note: In case of multi-box setup, run the above commands in the respective boxes.
2. Run the downgrade command.

   Note:
   - When you downgrade, the backup of the `site-options.conf` file will be available in `/opt/collabnet/teamforge-installer/7.0.0.0/conf` directory.
   - You must downgrade TeamForge based on the services enabled in the box in case of an advanced mode setup. For example, run the downgrade command `yum downgrade teamforge-scm CN-collabnet-core CN-migration CN-runtime TeamForge-installer` in case of "SCM on a separate box".

   ```
   yum downgrade CN-collabnet-core CN-runtime CN-migration teamforge
   TeamForge-installer teamforge-app teamforge-database teamforge-etl
   teamforge-scm teamforge-codesearch ctf_clireports
   ```

   If you have git integration, add `"teamforge-git"` to the `yum downgrade` command as follows:

   ```
   yum downgrade CN-collabnet-core CN-runtime CN-migration teamforge
   TeamForge-installer teamforge-app teamforge-database teamforge-etl
   teamforge-scm teamforge-codesearch teamforge-git ctf_clireports
   ```
3. Recreate the run time environment.

   ```
   cd /opt/collabnet/teamforge-installer/7.0.0.0
   ```

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

4. If you have Git integration, run the following command.

```
ln -sf /opt/collabnet/gerrit/bin/gerrit.sh /etc/init.d/gerrit
```

5. If your TeamForge site is running in advanced mode, start the external services by running the following commands.
   - Start the Apache server.

     ```
     /etc/init.d/httpd start
     ```
   - If the TeamForge instance uses PostgreSQL database, start the PostgreSQL.

     ```
     /etc/init.d/postgresql-9.0 start
     ```
6. Start TeamForge.

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

To verify if the patch has been reverted:

Click **Admin** in the site navigation bar.
On the site administration navigation bar, click **System Tools**.
Click **Build Information** and observe the following:

Version: 7.0.0.0
Applied Patches: None

Maintain your TeamForge 7.0 site

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.

Upgrade CLI reports to the latest version on Red Hat, CentOS or SUSE

Use these instructions to upgrade the CLI reports to the latest version.

Though life cycle metric reports are released as part of regular TeamForge product releases (including patch releases), new reports are, at times, made available to customers as and when they are ready. When a new report is available, you can upgrade your CLI reports to the latest version.

1. Log on to TeamForge application server as root.
2. Install the latest `ctf_clireports`.

   Run the following command for Red Hat/CentOS:

   ```
   # yum install ctf_clireports
   ```

   Run the following command for SUSE:

   ```
   # zypper install ctf_clireports
   ```
3. Run the `post-install.py` script.

   ```
   /opt/collabnet/teamforge/runtime/scripts/post-install.py
   ```
4. Run the following command and verify the CLI reports installation. Observe the version and release information.

   ```
   # rpm -qi ctf_clireports
   ```

Install Indexer on a separate server

In this task, we install the TeamForge Indexer on a separate server.

Pre-requisite: Ensure that there is no firewall restriction between the TeamForge application server and the new Indexer server.

Do this on the TeamForge app server.

1. Stop TeamForge.

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

2. Move the Indexer directory `[/opt/collabnet/teamforge/var/searchIndex]` to the new Indexer server.

Note: You can either use tar or NFS to backup and restore the searchIndex directory. Copy or transfer the searchIndex backup to the indexer server. See the example below.

```
cd /opt/collabnet/teamforge/var
tar -zcvf /tmp/searchIndex.tgz searchIndex
scp /tmp/searchIndex.tgz user@indexer.box:/tmp/
```

3. Edit the `site-options.conf` file.

```
HOST_my.host.name=app etl database datamart subversion cvs gerrit
```

```
HOST_<domain/hostname of indexer box>=indexer
```

4. Run the installer.

```
cd /var/ops/teamforge-installer/7.0.0.1
sudo ./install.sh -r -I -d /opt/collabnet/teamforge
```

5. Start TeamForge.

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

Do this on the TeamForge Indexer server.

6. Setup the server with an operating system supported by TeamForge 7.0
7. Copy the following repository to your production server.

```
sudo vi /etc/yum.repos.d/collabnet-7.0.0.1.repo
```

```
[CollabNet]
name=collabnet
baseurl=http://packages.collab.net/7.0.0.0/redhat/$releasever/$basearch
gpgkey=http://packages.collab.net/RPM-GPG-KEY-collabnet
enabled=1
gpgcheck=0
```

8. Run the following command to install the TeamForge Indexer packages.

```
yum install teamforge-indexer
```

9. Copy the master `site-options.conf` file from the application server and modify these tokens:

```
HOST_my.host.name=indexer
DOMAIN_my.host.name=<myindexerbox.domain.com>
HOST_<myappboxdomain.com>=app etl database datamart subversion cvs gerrit
```

Important:

- It is recommended to use a separate netapp volume for Indexer server.
- The Indexer server should have the required permissions to access the NFS volume of app server.

10. Create the Indexer and filestorage directories.

```
mkdir /opt/collabnet/teamforge/var/searchIndex -p
mkdir /opt/collabnet/teamforge/var/filestorage -p
```

11. Restore the Indexer data.

Run the following command to restore the backup index data from the app server.

```
cd /opt/collabnet/teamforge/var/
tar -zxvf /tmp/searchIndex.tgz
```

12. Run the installer.

```
cd /var/ops/teamforge-installer/7.0.0.1
sudo ./install.sh -r -I -d /opt/collabnet/teamforge
```

13. Start the TeamForge Indexer.

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

a) Run the following command and ensure that app filestorage directory is exported to indexer server.

```
mount -l
```

b) Run the following command and ensure that the Indexer services are up and running.

```
sudo /etc/init.d/collabnet status
```

Supply your TeamForge license key from Teamforge user interface

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 234.

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:
   `25:supervillaininc:144.16.116.25.:302D02150080D7853DB3E5C6F67EABC65BD3AC17D4D35CB3Z002`

   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:
`alm=100:SCM=80:supervillaininc:144.16.116.25.:302D02150080D7853DB3E5C6F67EABC65BD3AC17`

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 0664 /opt/collabnet/teamforge/var/etc/sflicense.txt
chown <APP_USER>:<APP_GROUP> /opt/collabnet/teamforge/var/etc/
sflicense.txt
```

   Note: Change the values of `<APP_USER>` and `<APP_GROUP>` with the values of *APP_USER* and *APP_GROUP* tokens respectively from the `/opt/collabnet/teamforge/runtime/conf/runtime-options.conf` file.

Support CollabNet TeamForge System Administrators

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: Refer to the *Software requirements for TeamForge 7.0* topic for the supported OpenLDAP versions.

1. Shut down CollabNet TeamForge .

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

2. 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) Configure the following tokens.

      Important: The values specified for the following tokens are only for illustration purpose.

```
EXTERNAL_AUTHENTICATION_TYPE=ldap
LDAP_DN_PREFIX=cn=
LDAP_DN_SUFFIX=,cn=Users,dc=testldap,dc=qa,dc=collab,dc=net
LDAP_SERVER_URL=ldap://testldap.qa.collab.net:3268
```

3. Recreate the runtime environment.

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

4. Start TeamForge.

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

Modify the application policy

To enable CollabNet TeamForge to authenticate against your LDAP server, modify the application-policy block of the `standalone-full.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/7.0.0.1/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 from where you can quickly retrieve it to your TeamForge 7.0 site.

- If you are upgrading by installing TeamForge 7.0 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/apache2 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/9.0/backups/
teamforge_data_backup.dmp
exit
mkdir /tmp/backup_dir
cp /var/lib/pgsql/9.0/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/9.0/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 62backup.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 TeamForge data, set up the Oracle database and tell the installer how to handle it.

TeamForge database setup

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 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> grant create any synonym to <sf user>;
   SQL> exit
   ```
5. Create the database read-only user that you will use to connect from CollabNet TeamForge.

   ```
   SQL> create user <database_readonly_user> identified by
   <database_readonly_password> default tablespace <your tablespace>
   temporary tablespace <temporary tablespace>;
   ```
6. Grant the required permissions to the read-only user that you just created.

   ```
   SQL> grant create session to <database_readonly_user>;
   SQL> exit
   ```

Datamart setup

7. 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*.
8. Connect to your Oracle database.

   ```
   SQL> connect <adminusername>@<db_name>/<adminpassword> as sysdba
   ```
9. Create the datamart user you will use to connect from CollabNet TeamForge.

   ```
   SQL> create user <reports_database_user> identified by
   <reports_database_password> default tablespace <your tablespace> temporary
   tablespace <temporary tablespace>;
   ```

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

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

Note: Replace <reports_database_user> with the datamart username specified in the `site-options.conf` and <reports_database_password< with the database password specified in site-options.conf.

11. Create the datamart read-only user that you will use to connect from CollabNet TeamForge.

```
SQL> create user <reports_readonly_user> identified by
<reports_readonly_password> default tablespace <your tablespace> temporary
tablespace <temporary tablespace>;
```

12. Grant the required permissions to the read-only user that you just created.

```
SQL> grant create session to <reports_readonly_user>; SQL> exit
```

Note: 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 index content change for the search functionality in TeamForge 6.1.1, a complete re-indexing of the site is required if you are upgrading from TeamForge 6.1 or earlier versions of TeamForge. 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/7.0.0.1/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/7.0.0.1/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/7.0.0.1/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/7.0.0.1/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/7.0.0.1/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/7.0.0.1/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/7.0.0.1/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 manually

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/7.0.0.1
./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.

Install project templates using the installer

With TeamForge 7.0, you can install project templates automatically.

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.

- To install project templates automatically when the TeamForge startup script is executed, you must set these tokens:
  - `INSTALL_TEMPLATES = true`
  - `REQUIRE_USER_PASSWORD_CHANGE = false`

  Note:

  - The TeamForge startup script installs the sample TeamForge project templates by default.
  - If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
  - You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

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/7.0.0.1/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 257.

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/7.0.0.1/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:

   Note: In case of a multibox setup, modfiy the token *PERFORCE_PORT=0.0.0.0:1666* in the `site-options.conf` file in all the boxes where the service is distributed to get the correct status of the Perforce server.

   ```
   PERFORCE_PORT=localhost:1666
   PERFORCE_CLIENT=/usr/local/bin/p4
   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 258.

Protect your CollabNet TeamForge site

You can take various measures to maximize the security of your CollabNet TeamForge users.

Encrypt communication between the application and database servers

To prevent your data from being exposed in a readable format on the network, use Secure Socket Layer (SSL) to encrypt the network traffic between the application and the database servers.

If you have a dedicated server for your database (operational database or datamart database), encrypt the data traffic between the application and database servers and between the ETL and datamart servers.

👉 **Important:** The following steps are applicable only in a dedicated multi-box installation setup.

1. Stop TeamForge in all the servers.

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

2. Add the following site option tokens in all the TeamForge servers.
   a) If the operational database is running on a separate server, set the token *DATABASE_SSL=on*.
   b) If the datamart database is running on a separate server, set the token*REPORTS_DATABASE_SSL=on*.

   👉 **Note:** It is mandatory to include the tokens specified above in all the servers.

3. Re-create the runtime environment in all the TeamForge servers.

   ```
   cd /opt/collabnet/teamforge-installer/7.0.0.1
   ./install.sh -r -n -V -d /opt/collabnet/teamforge
   ```

4. Start TeamForge in all the servers.

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

5. Verify that your Postgresql database is running in the SSL mode.
   a) Log in to the database server.
   b) Run the following command:

      ```
      sudo grep "ssl = " /var/lib/pgsql/9.0/data/postgresql.conf
      ```

      Observe:"ssl = on"

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.

Pre-requisites to setup the SELINUX environment:

- TeamForge SElinux can be configured only for the TeamForge application server.
- TeamForge SELINUX supports RHEL/CentOS 6.4 and higher versions.
- In case of same box upgrade using RHEL/CentOS, the OS upgrade is recommended to 6.4 and higher versions.
- TeamForge SELINUX is not supported for customers migrating from CEE to CTF.

1. Stop TeamForge in all the servers.

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

2. Setup SELINUX in enforcing mode in the TeamForge application server.
   a) Edit the file `/etc/sysconfig/selinux` and set the parameter.

      ```
      SELINUX=enforcing
      ```

   b) Turn off TeamForge startup on boot.

      ```
      chkconfig collabnet off
      ```

   c) Reboot the server.
   d) Ensure that SELINUX is running in enforce mode.

```
getenforce
```

3. Add the following site option token in the TeamForge application server.

```
SELINUX_SETUP=true
```

4. Run the following commands to enable the publishing repository commits in the SELINUX mode.

```
sudo rm -f /tmp/remotePublishLog.txt
sudo su - <HTTPD_USER> -c "touch /tmp/remotePublishLog.txt"
sudo semanage fcontext -a -t httpd_sys_rw_content_t '/tmp/
remotePublishLog.txt'
sudo restorecon -v /tmp/remotePublishLog.txt
```

Run the following commands if the `SSL_CHAIN_FILE` is configured.

```
sudo semanage fcontext -a -t cert_t '<SSL_CHAIN_FILE>'
sudo restorecon -v <SSL_CHAIN_FILE>
```

Run the following commands if the `SSL_CA_CERT_FILE` is configured.

```
sudo semanage fcontext -a -t cert_t '<SSL_CA_CERT_FILE>'
sudo restorecon -v <SSL_CA_CERT_FILE>
```

Important: Get the value of the <HTTPD_USER> from the `runtime-options.conf` file.

5. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

6. Fix the SELINUX data permissions.

```
cd /opt/collabnet/teamforge/runtime/scripts
./fix_data_selinux_permissions.sh
```

Note: If your data directory is on a NetApp volume, you may observe some warning messages which can be ignored safely.

7. Run the following commands if Review Board is integrated with TeamForge.

```
sudo semanage fcontext -a -t httpd_sys_rw_content_t "/opt/collabnet/
reviewboard/data(/.*)?"
sudo restorecon -R -v /opt/collabnet/reviewboard/data
sudo semanage fcontext -a -t httpd_sys_rw_content_t "/u1/
reviewboard(/.*)?"
sudo restorecon -R -v /u1/reviewboard
semanage fcontext -a -t httpd_sys_rw_content_t "/opt/collabnet/teamforge/
var/home/apache(/.*)?"
restorecon -R -v /opt/collabnet/teamforge/var/home/apache
```

8. Start TeamForge.

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

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/apache2/httpd.conf` `/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/7.0.0.1/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/apache2/conf.d/ssl.conf` `/etc/httpd/conf.d/ssl.conf` file to `/etc/apache2/conf.d/ssl.conf.old` `/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/apache2 stop
   /etc/init.d/postgresql-9.0 stop /etc/init.d/postgresql stop
   /etc/init.d/collabnet stop
   ```

8. Run the installer.

   ```
   cd /opt/collabnet/teamforge-installer/7.0.0.1
   ./install.sh -r -I -V
   ```

9. Rename the `/etc/apache2/httpd.conf.cn_new` `/etc/httpd/conf/httpd.conf.cn_new` file to `httpd.conf`, if it exists.

   Also rename the `/etc/sysconfig/apache2.cn_new` file to `apache2`, if it exists.
10. Start TeamForge.

    ```
    /etc/init.d/httpd start
    /etc/init.d/httpd start
    /etc/init.d/apache2 start
    /etc/init.d/postgresql-9.0 start /etc/init.d/postgresql 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.

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*.

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/7.0.0.1/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:

   Note: If you want to run the ETL and datamart services on a separate box, see *Install TeamForge the advanced way* for details.

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
   REPORTS_DATABASE_MAX_POOL_SIZE=30
   ```

4. If you want your datamart to use a different port than your site database, we recommend the following:

   - For small dataset customers: use the same instance as that of `ctfdb`; leave REPORTS_DATABASE_PORT commented.

- For medium to large dataset customers: use a separate instance by setting REPORTS_DATABASE_PORT=5632.

5. For ETL to function, 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 350.

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 361 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 CVS 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 on RedHat or CentOS

Use the yum package manager to upgrade to the latest supported Subversion release.

TeamForge 7.0 supports Subversion 1.7.11.

1. Log in as `root` and stop all `collabnet` services.
2. Check that Subversion 1.7.11 is available for upgrade:

   ```
   yum list subversion
   ```
3. Upgrade Subversion.

   ```
   yum update subversion subversion-perl subversion-python mod_dav_svn
   ```
4. Verify the Subversion upgrade.

   ```
   rpm -qa | grep subversion
   svn --version | grep "version"
   ```
5. Change to the `runtime/scripts` directory.

   ```
   cd /opt/collabnet/teamforge/runtime/scripts
   ```
6. Start all `collabnet` services.

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

Upgrade Subversion on SuSE

Use the zypper package manager to upgrade to the latest supported Subversion release.

TeamForge 7.0 supports Subversion 1.7.11.

1. Login as `root` and stop all `collabnet` services.
2. Check that Subversion 1.7.11 is available for upgrade:

   ```
   zypper refresh
   zypper repos
   ```
3. Remove the older Subversion packages.

   ```
   zypper remove subversion subversion-server subversion-tools subversion-
   perl subversion-python
   ```
4. Upgrade Subversion.

   ```
   zypper install subversion subversion-tools subversion-perl subversion-
   python
   zypper install subversion-server
   ```

   Note: You may see a message like this: "There are some running programs that use files deleted by recent upgrade. You may wish to restart some of them. Run 'zypper ps' to list these programs". This message appears during a package upgrade which might cause library files to be overwritten -- if there any running processes using older library files, zypper warns you to restart those processes. You can ignore this message.
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
   ```
7. Start all `collabnet` services.

```
/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 such as the Project Tracker or the Review Board. 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.py --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
   ```
7. If your site has Review Board integrated, run the following commands:
   a) Start the TeamForge application services.

      ```
      sudo /etc/init.d/collabnet start all
      ```
   b) Go to the RBInstaller-1.3.0.0.13 directory.

      ```
      cd /var/ops/RBInstaller-1.3.0.0.13
      ```
   c) Update the `install.conf` file with the new domain name.

      ```
      vi /var/ops/RBInstaller-1.3.0.0.13/installer/install.conf
      ```
   d) Recreate the runtime on Review Board.

      ```
      sudo python ./install.py -r
      ```
   e) Restart the TeamForge application services.

      ```
      sudo /etc/init.d/collabnet restart all
      ```

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/7.0.0.1/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

Set the logging level appropriately to enable logging in `vamessages.log` and in `James` logs.

Follow these instructions to enable debug logging only in `vamessages.log`.

1. Edit the `$RUNTIME/jboss/standalone/configuration/standalone-full.xml` file to change the logging level in `vamessages.log`.
2. Locate the following lines in the `standalone-full.xml` file.

```
<logger category="com.vasoftware" use-parent-handlers="false">
<level name="INFO" />
<handlers>
```

```
<handler name="VAFILE" />
</handlers>
</logger>
```

3. To change the log level from "INFO" to "DEBUG" do the following.

```
<logger category="com.vasoftware" use-parent-handlers="false">
<level name="DEBUG" />
<handlers>
<handler name="VAFILE" />
</handlers>
</logger>
```

4. Restart the JBoss service to pick up these changes.

```
/etc/init.d/collabnet restart jboss
```

Note: These changes will not be available next time when you recreate the runtime environment.

Note: To enable `James` logs such as `Mailet`, `James`, `SpoolManager` and so on, follow these steps:

- Edit `$RUNTIME/james/apps/james/SAR-INF/environment.xml` .
- Locate "categories" and set the log-level to DEBUG for the categories in which you wish to enable logging.
- Restart `James`.

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/apache2 stop
/etc/init.d/postgresql-9.0 stop /etc/init.d/postgresql 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/7.0.0.1/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/7.0.0.1
```

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

5. Start TeamForge.

```
/etc/init.d/httpd start
/etc/init.d/httpd start
/etc/init.d/apache2 start
/etc/init.d/postgresql-9.0 start /etc/init.d/postgresql start
/etc/init.d/collabnet start
```

Rotate TeamForge log files

To keep log files under control, move them to a compressed archive.

Rotate logs automatically

Set up your site to move log files automatically to a compressed archive every day.

1. Check that the system's cron jobs are set.

```
sudo crontab -l
```

If the cron jobs are in place, you get output like this:

```
# Begin sourceforge cron jobs (Generated by the build.  DO NOT HAND EDIT.)
 --
4,9,14,19,24,29,34,39,44,49,54,59 * * * * /opt/collabnet/teamforge/
runtime/scripts/run-groups.sh 5mins
0 * * * * /opt/collabnet/teamforge/runtime/scripts/run-groups.sh hourly
0 0 * * * /opt/collabnet/teamforge/runtime/scripts/run-groups.sh daily
0 0 * * 0 /opt/collabnet/teamforge/runtime/scripts/run-groups.sh weekly
0 0 1 * * /opt/collabnet/teamforge/runtime/scripts/run-groups.sh monthly
# End sourceforge cron entries
 ----------------------------------------
* * * * * /usr/bin/aiupdate -n -d http://mgr/dist >/dev/null 2>&1
```

If the output does not appear, you need to set up the cron jobs.

2. In the `/opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf` file, set the value of the *INSTALL_CRON* option to true.
3. Recreate the runtime environment.

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

All your site's logs will now be automatically rotated into a compressed archive every 24 hours.

Rotate logs manually

Use the `rotatelogs.pl` script to move log files to a compressed archive.

Do this when a log file has become so large it is difficult to use.

Run the `rotatelogs.pl` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/rotatelogs.pl
```

All your site's logs are moved into a compressed archive and new log files are started.

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/7.0.0.1/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.

Removing users from monitoring objects

As a site or project administrator, if one or more users are no longer project members, you can remove them from monitoring selected TeamForge objects they once subscribed for monitoring.

However, you cannot remove a user from the monitoring list if the user is *monitoring applications* such as trackers, documents, tasks, and so on instead of individual TeamForge objects.

By default, this feature is disabled. To enable this feature, set the *USER_MONITORING_REMOVE_ENABLED* variable to true in the `site-options.conf` file.

Note: Every user removal operation is being logged in the database for audit purposes.

1. Go to the item, from which you want to remove users from monitoring.
2. Select users to remove from monitoring list.
   a) If you want to remove one or more users from monitoring one of the items, select the item, then click **Monitor** > **Users Monitoring Selected**.
   The **Users Monitoring This Item** window appears.
   b) If you want to remove one or more users from monitoring more than one item, select all the items, then click **Monitor** > **Users Monitoring This Folder**.
   The **Users Monitoring This Folder** window appears.
   c) Select one or more check boxes corresponding to the users you want to remove from monitoring.
3. Click **Remove**.
   The `Are you sure you want to remove the selected user(s) from monitoring?` message appears.
4. Click **OK**.

The selected users are removed from monitoring the selected object. An e-mail notification is sent to all active users that are removed from monitoring selected objects.

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/apache2 start
   /etc/init.d/postgresql 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

To back up a PostgreSQL database, use the `db.pl` script or the `pg_dump` command depending on your setup.

Note: You can use `db.py` to back up PostgreSQL data in all setups except the dedicated database server setup where `ctf` (TeamForge database) is not installed. In a dedicated database setup, you can back up the PostgreSQL database safely while it is online by using the native `pg_dump` command.

Using `db.py`:

- To back up the entire PostgreSQL database, run this command.

  ```
  ./db.py -a dump -f /tmp/backup/
  ```

- To back up only the `ctf` database, run this command.

  ```
  ./db.py -a dump -t ctf -f /tmp/backup/
  ```

- To back up only the datamart, run this command.

  ```
  ./db.py -a dump -t reporting -f /tmp/backup/
  ```

Using `pg_dump`:

- In a dedicated database setup, you can run one of these `pg_dump` commands:

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

- `pg_dump -Ft -b -o ctfdb > ctfdb.tar`
- `pg_dump -Fc <dbname> -f ctfdb.dmp`
- `pg_dump ctfdb > /tmp/backup_dir/teamforge_data_backup.dmp`

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` or `psql` commands, or using the `db.py` script.

Note: You can use `db.py` to restore PostgreSQL data in all setups except the dedicated database server setup where dedicated database server setup where `ctf` (TeamForge database) is not installed. In a dedicated database setup, you can restore the PostgreSQL database using the `pg_restore` or `psql` command.

Using `db.py`:

- To restore the entire PostgreSQL database, run this command.

  ```
  ./db.py -a restore -f /tmp/backup/
  ```

- To restore only the `ctf` database, run this command.

  ```
  ./db.py -a restore -t ctf -f /tmp/backup/
  ```

- To restore only the datamart, run this command.

  ```
  ./db.py -a restore -t reporting -f /tmp/backup/
  ```

Using `pg_restore` or `psql`:

- In a dedicated database setup, you can use `pg_restore` or `psql`:
  a) Locate the dump file you created when backing up the PostgreSQL database.
  b) Shut down CollabNet TeamForge .
  c) Create a database and user with the names used for CollabNet TeamForge .

  ```
  createuser -U $CTFUSER createdb -E UNICODE -U $CTFUSER ctfdb
  ```

  d) Restore the database using either the `pg_restore` or `psql` utility. Run one of these commands:

  - `pg_restore -d ctfdb ctfdb.tar`
  - `pg_restore -d <username> -d <dbname> ctfdb.dmp`
  - `psql ctfdb < /tmp/backup_dir/teamforge_data_backup.dmp`

  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.

Move the datamart to a separate box

TeamForge 6.2 supports a multi-box setup in all modes except for the dedicated database server mode. The setup with all services on one box includes having the datamart in the same PostgreSQL instance as TeamForge as well as running it in a separate instance. In either case, you can now move the datamart to a separate box.

Move the datamart (dedicated database server mode)

In this task, we move the PostgreSQL datamart from its own instance to a separate box in the dedicated database server mode.

1. Stop TeamForge on the `app` box.

   If this is a multi-box scenario, stop TeamForge on all other boxes as well.

   ```
   [RUNTIME_DIR]/scripts/collabnet stop
   ```

2. Do a dump of the PostgreSQL datamart. `su - postgres -c 'pg_dump -C -p <database-port > <reports-database-name > <path-to-dump-file>`
3. Create a new datamart instance using the `datamart-pgsql-setup.sh` or follow the instructions below in the database box.

   ```
   su - postgres
   ```

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

4. Set the `REPORTS_DATABASE_PORT` in `site-options.conf`

   Note: The port should use the same value as specified in `postgresql.conf` as specified in the previous step. The recommended value is 5632.

5. Re-create the [RUNTIME_DIR]in all the boxes.

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

6. Restore the datamart into the new instance.

   ```
   [RUNTIME_DIR]/scripts/db.py -a restore -t reporting -f <dump-location>
   ```

7. Copy the postgresql-9.0 script from runtime scripts and replace `/etc/init.d/postgresql-9.0`
8. Restore the datamart from the database box.

   ```
   su - postgres -c 'psql -p <reports-database-port> <path-to-dump-file>
   ```

   Note: Restart the Postgres service. If any warning messages are displayed, kill the service and start again.

9. Start the database in the database box.

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

10. Start the services in all boxes.

    ```
    [RUNTIME_DIR]/scripts/collabnet start all
    ```

11. Check if the existing data appears in charts.
12. Permanently remove the old datamart from the TeamForge instance.

    ```
    su - postgres -c 'dropdb <datamart-name> -p <database-port>'
    ```

Move the datamart (other modes)

In this task, we move the PostgreSQL datamart from its own instance to a separate box in all other modes, except dedicated database server mode.

1. Stop TeamForge on the `app` box.

   If this is a multi-box scenario, stop TeamForge on all other boxes as well.

   ```
   [RUNTIME_DIR]/scripts/collabnet stop
   ```

2. Do a dump of the PostgreSQL datamart. `[RUNTIME_DIR]/scripts/db.py -a dump -t reporting -f <dump-location>`
3. Create a new datamart instance using `datamart-pgsql-setup.sh` or follow the instructions below in the database box.

   ```
   su - postgres
   ```

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

4. Set the `REPORTS_DATABASE_PORT` in `site-options.conf`

   Note: This value should be different from the value of DATABASE_PORT. The recommended value is 5632.

5. Re-create the [RUNTIME_DIR]

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

6. Restore the datamart into the new instance.

   ```
   [RUNTIME_DIR]/scripts/db.py -a restore -t reporting -f <dump-location>
   ```

7. Start the services in all boxes.

   ```
   [RUNTIME_DIR]/scripts/collabnet start all
   ```

8. Check if the existing data appears in charts.
9. Permanently remove the old datamart from the TeamForge instance.

   ```
   su - postgres -c 'dropdb <datamart-name> -p <database-port>'
   ```

Integrate TeamForge 7.0 with other tools

TeamForge 7.0 supports integrations with third-party tools for versioning, reviewing and searching source code.

Integrated application example: Pebble

Pebble is a blogging application that has been enhanced to support quick and easy integration with TeamForge.

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.

Option	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 262 for how to use them.

If you have installed Pebble with SSL, restart the TeamForge server before integrating Pebble with TeamForge. You can restart the TeamForge server using `/etc/init.d/collabnet restart all`

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 261.

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.

Set up Black Duck Code Sight

TeamForge 7.0 supports the Black Duck Code Sight source code search engine.

To install Black Duck Code Sight, see the instructions specific to your platform.

Install TeamForge 7.0 with Black Duck Code Sight on a separate box on Red Hat/CentOS

In this option, we install Black Duck Code Sight on a separate box on Red Hat Enterprise Linux and other services on the main application box.

Important: There are no changes in the steps for installing TeamForge 7.0 Patch 1

Note: In a multi-box installation of TeamForge, all machines must have the same time zone - without this, ETL will not function properly.

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

1. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.
   - The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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 7 for details.
3. Set the TeamForge repository configuration.

   If you are installing TeamForge with internet access:
   - Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
   - Copy it to `/etc/yum.repos.d/`.
   - Refresh your repository cache.

     ```
     yum clean all
     ```

   If you are installing TeamForge without internet access:

   a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

   b) Unpack the disconnected installation package.

   ```
   rpm -ivh <package-name>
   ```

   c) **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

   Insert the Red Hat installation DVD.

   This will provide some of the necessary software and utilities for installing TeamForge.

   d) Mount the Red Hat installation DVD.

   ```
   cd /media/
   mkdir cdrom
   mount /dev/cdrom ./cdrom/
   ```

   Replace `cdrom` with the identifier for your server's CD drive, if necessary.

   Tip: If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

   e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

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

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

Tip: If the yum installer fails, you may have duplicate rpm packages installed.

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 teamforge-scm
```

5. Install the following application packages.

a) GIT: To install the GIT packages run the following command.

```
yum install teamforge-git
```

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

```
vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
```

At this stage, we'll make the minimal modifications required to set up a usable TeamForge 7.0 site.

Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

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) Configure the TeamForge services to run in the local box.

```
HOST_localhost=app etl database datamart cvs subversion indexer
```

c) Review the *DATABASE* variables.

You may want to substitute your own values for the default database names, user names and passwords.

d) Turn on the site-wide reporting functionality by adding the *REPORTS_** variables.

See *Turn on site-wide reporting* on page 250 for details.

e) By default, the site wide reports are not commented. To turn on the site wide reports, set the values. Else, comment out the following reports.

- REPORTS_DATABASE_NAME=
- REPORTS_DATABASE_PASSWORD=
- REPORTS_DATABASE_USERNAME=
- REPORTS_DATABASE_READ_ONLY_USER=
- REPORTS_DATABASE_READ_ONLY_PASSWORD=

f) **It is mandatory to include the *SCM_DEFAULT_SHARED_SECRET* token in the `site-options.conf` file of the primary TeamForge server, and provide it with a value of 16-24 characters. Remember to use the same key in the external SCM integration server also.**

g) If the token *REQUIRE_RANDOM_ADMIN_PASSWORD* is already set to true, then set the token *ADMIN_EMAIL* with a valid email address.

```
ADMIN_EMAIL=root@{__APPLICATION_HOST__}
```

h) 👉 **Important:** If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

Comment out the token if site-wide reports is not enabled.

i) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

j) Configure the following settings if you are installing Git:

```
HOST_localhost=app etl database datamart subversion cvs indexer gerrit
```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

k) Configure the following settings for Black Duck Code Sight.

```
HOST_my.codesight.hostname=codesearch
```

👉 **Note:** In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>

To enable SSL for Black Duck Code Sight, include this token:
BDCS_SSL=on
```

👉 **Note:** If you have Black Duck Code Sight on SSL and the following external certificate tokens are not provided, the installer will generate a self-signed certificate.

```
For valid SSL certificates, configure the following tokens:
BDCS_SSL_CERT_FILE=
BDCS_SSL_KEY_FILE=
The ca.crt and chain files are optional -- leave out the tokens if you
 don't use the files.
BDCS_SSL_CA_CERT_FILE=
BDCS_SSL_CHAIN_FILE=
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
BDCS_ADMIN_PASSWORD=<password>
```

```
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

Advanced Black Duck Code Sight configuration settings:

Note: The following are the advanced configuration tokens which can be set once during the time of installation.

```
The path where the repositories enabled for codesearch to check out.
BDCS_SCAN_SOURCE_DIR_ROOT=/opt/collabnet/blackduck/scan

The path where the codesearch software is installed.
BDCS_INSTALL_PATH=/opt/collabnet/blackduck

The path where codesearch database is installed.
BDCS_PGSQL_HOME_DIR_ROOT=/opt/collabnet/blackduck/postgres

The port number for codesearch db server.
BDCS_PGSQL_PORT=55435

The tomcat maximum heap memory size in megabytes.
BDCS_TOMCAT_MX_IN_MB=1024

The shutdown port number for codesearch tomcat server.
BDCS_TOMCAT_SHUTDOWN_PORT=9189
```

7. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

Note: If you are setting up CollabNet TeamForge on a machine that is meant for emergency backup or disaster recovery, use the -R option in place of the -r option.

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

```
export http_proxy=
```

Do this on the Code Sight server. We'll call this my.codesight.box

9. Install Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.
   - The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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. Run the following command to install Black Duck Code Sight.

```
yum install teamforge-codesearch
```

11. Copy the `site-options.conf` file from the application box to the code search box in the directory `/opt/collabnet/teamforge-installer/7.0.0.1/conf`
12. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

Do this on my.app.box:

13. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./bootstrap-data.sh
```

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.

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

15. Install the Black Duck Code Sight license on the box where Black Duck Code Sight is installed. For more information, see *Install the Black Duck Code Sight license* on page 274.

16. Integrate gerrit by running the `post-install.py` script.

Tip: Skip this step if you are not installing git integration.

```
/opt/collabnet/gerrit/scripts/post-install.py
```

The post installation script will try to detect the required configuration parameters. The following 3 parameters do not have default values and you will be asked to provide them:

- Teamforge Login name: the dedicated TeamForge site administrator account that does not expire and cannot be locked
- Teamforge Password: the password for the above account
- Database password: the password to protect Gerrit's database from unauthorized access. Specify its value when you first run the post-install script. Make sure you note the value because you will be asked for it later.

a) Restart the Gerrit services.

```
/etc/init.d/collabnet restart gerrit
```

b) To verify the Git integration, run the following command.

```
/etc/init.d/collabnet status
```

17. Integrate the CLI reports by running the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

18. Run the following initial ETL load jobs.

a) Change to the `runtime/scripts` directory.

```
cd /opt/collabnet/teamforge/runtime/scripts
```

b) Run the `TrackerInitialJob`.

```
./etl-client.py -r TrackerInitialJob
```

c) Run the `SCMInitialJob`.

```
./etl-client.py -r SCMCommitInitialJob
```

Tip: For more information see *When do I run the initial load job?* on page 294.

Do this in my.codesight.box

19. To integrate Black Duck Code Sight with TeamForge run the Black Duck Code Sight `post-install.sh` script.

    Note: It is assumed that Subversion's client configuration file (`/root/.subversion/config`) for the root user is the default one without customization.

    ```
    sudo /opt/collabnet/teamforge/runtime/scripts/codesearch/post-install.sh
    ```

20. If the token *VALIDATE_SSL_CERTS* is set to "true", you must run the codesearch runtime script *trust-cert.sh* in the application box.

    ```
    cd /opt/collabnet/teamforge/runtime/scripts/codesearch
    ./trust-cert.sh
    ```

Do this on my.app.box

21. Restart the TeamForge services.

    ```
    /etc/init.d/collabnet restart tomcatcs
    ```

Do this on my.codesight.box

22. Restart the Black Duck Code Sight services.

    ```
    /etc/init.d/collabnet restart
    ```

23. Apply some finishing touches and make sure everything is running smoothly.
    a) 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.
    b) Create a sample project.

       See *Create a TeamForge project*.
    c) Write a welcome message to your site's users.

       See *Create a site-wide broadcast*.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Install Black Duck Code Sight on SuSE

In this option, we install Black Duck Code Sight on a separate box on SuSE and other services on the main application box.

Important: There are no changes in the steps for installing TeamForge 7.0 Patch 1

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

1. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.
   - See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
   - See *the SuSE Linux Enterprise Server deployment 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 7 for details.
3. Set up the SuSE repository.
   a) Register the machine with Novell.

      ```
      suse_register -i -a email=<MailID> -a regcode-sles=<Registration code>
      ```

where <MailID> is the email address associated with your SuSE Linux Enterprise Server license and <Registration code> is the Suse Enterprise Linux license code.

b) Move the existing installer repository out of the way.

```
cd /etc/zypp/repos.d/
mv SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo_old
```

c) Make sure the correct Novell repository is enabled.

Open `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo` in a text editor and confirm that `enabled=1`.

d) Refresh `zypper`.

```
zypper ref
```

Note: As an alternative, you can insert the SuSe Linux Enterprise Server DVD.

- Ensure that the DVD is mounted in the location specified by the *baseurl* variable in `SUSE-Linux-Enterprise-Server-11-SP2 11.2.2-1.234.repo`.
- Use `zypper info postgresql` to check that the DVD is properly mounted.

4. If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
zypper ref
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
SuSE Linux Enterprise Server 11 SP2 32-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.i686.rpm`
SuSE Linux Enterprise Server 11 SP2 64-bit	`CTF-Disconnected-media-7.0.0.1-380.sles11.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) **Note:** If you've already set up the SuSE repository earlier, skip the DVD instructions in this step and the next.

Insert the SuSE installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) To check that the DVD is mounted properly, run the command `zypper info postgresql`. It must be mounted in the location specified in the *baseurl* variable in `/etc/zypp/repos.d/SUSE-Linux-Enterprise-Server-11-SP1\ 11.1.1-1.152.repo`.

If necessary, replace `sr0` with the identifier for your server's CD drive.

Now the package manager will look for the SuSE installation components it needs on the DVD and not on the internet.

5. Install the following TeamForge application, database and other source control services.

```
zypper install teamforge-app teamforge-database teamforge-scm
```

6. Install the following application packages.

a) GIT: To install the GIT packages run the following command.

```
zypper install teamforge-git
```

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

   ```
   vi /opt/collabnet/teamforge-installer/7.0.0.1/conf/site-options.conf
   ```

   At this stage, we'll make the minimal modifications required to set up a usable TeamForge 7.0 site.

   Important: We noticed stability issues with the JVM option `-XX:+TieredCompilation` which is enabled by default in the JBoss's Java virtual machine. It is mandatory to turn off this option. To turn off, add `-XX:-TieredCompilation` to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

   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) Identify the boxes and the services running on them.

   ```
   HOST_localhost=app database subversion cvs
   DOMAIN_localhost=worlddomination.supervillain.org
   HOST_<my.datamartetlbox>=datamart etl
   ```

   c) Review the *DATABASE* variables.

   You may want to substitute your own values for the default database names, user names and passwords.

   d) Turn on the site-wide reporting functionality by adding the *REPORTS_** variables.

   See *Turn on site-wide reporting* on page 250 for details.

   e) By default, the site wide reports are not commented. To turn on the site wide reports, set the values. Else, comment out the following reports.

   - REPORTS_DATABASE_NAME=
   - REPORTS_DATABASE_PASSWORD=
   - REPORTS_DATABASE_USERNAME=
   - REPORTS_DATABASE_READ_ONLY_USER=
   - REPORTS_DATABASE_READ_ONLY_PASSWORD=

   f) **It is mandatory to include the *SCM_DEFAULT_SHARED_SECRET* token in the `site-options.conf` file of the primary TeamForge server, and provide it with a value of 16-24 characters. Remember to use the same key in the external SCM integration server also.**

   g) If the token *REQUIRE_RANDOM_ADMIN_PASSWORD* is already set to true, then set the token *ADMIN_EMAIL* with a valid email address.

   ```
   ADMIN_EMAIL=root@{__APPLICATION_HOST__}
   ```

   h)

   Important: If you have enabled site-wide reports, it is mandatory to set the *ETL_SOAP_SHARED_SECRET* token in the `site-options.conf` file of the primaryTeamForge server, and give it a value of 16-24 characters.

   Comment out the token if site-wide reports is not enabled.

   i) If you have LDAP set up for external authentication, you must set the "*REQUIRE_USER_PASSWORD_CHANGE*" site options token to false.

   j) Configure the following settings for Git:

   ```
   HOST_my.host.name=app etl database datamart subversion cvs
    indexer gerrit
   ```

To enable the history protection feature of TeamForge Git integration, set the **GERRIT_FORCE_HISTORY_PROTECTION=true**. For more info see *GERRIT_FORCE_HISTORY_PROTECTION* on page 362

k) Configure the following settings for Black Duck Code Sight.

```
HOST_localhost=app etl database datamart subversion cvs
 indexer codesearch
```

👉 **Note:** In case the HOST_ token is configured as HOST_localhost, then specify the following token with a valid hostname or domain name.

```
BDCS_HOST=<my.host.name or my.domain.name>

To enable SSL for Black Duck Code Sight, include this token:
BDCS_SSL=on
```

👉 **Note:** If you have Black Duck Code Sight on SSL and the following external certificate tokens are not provided, the installer will generate a self-signed certificate.

```
For valid SSL certificates, configure the following tokens:
BDCS_SSL_CERT_FILE=
BDCS_SSL_KEY_FILE=
The ca.crt and chain files are optional -- leave out the tokens if you
 don't use the files.
BDCS_SSL_CA_CERT_FILE=
BDCS_SSL_CHAIN_FILE=
```

```
To change the default Black Duck Code Sight admin password, add these
 tokens:
BDCS_ADMIN_USERNAME=sysadmin
BDCS_ADMIN_PASSWORD=<password>
To configure the port number for the Code Search Tomcat server, set this
 token:
BDCS_TOMCAT_PORT=9180
To specify the maximum results shown in Code Search, set this token:
Caution: Increasing this might impact performance.
BDCS_SDK_SEARCH_LIMIT_MAX=200
```

Advanced Black Duck Code Sight configuration settings:

👉 **Note:** The following are the advanced configuration tokens which can be set once during the time of installation.

```


The path where the repositories enabled for codesearch to check out.
BDCS_SCAN_SOURCE_DIR_ROOT=/opt/collabnet/blackduck/scan

The path where the codesearch software is installed.
BDCS_INSTALL_PATH=/opt/collabnet/blackduck

The path where codesearch database is installed.
BDCS_PGSQL_HOME_DIR_ROOT=/opt/collabnet/blackduck/postgres

The port number for codesearch db server.
BDCS_PGSQL_PORT=55435

The tomcat maximum heap memory size in megabytes.
BDCS_TOMCAT_MX_IN_MB=1024

The shutdown port number for codesearch tomcat server.
```

```
BDCS_TOMCAT_SHUTDOWN_PORT=9189
```

8. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

Note: If you are setting up CollabNet TeamForge on a machine that is meant for emergency backup or disaster recovery, use the -R option in place of the -r option.

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 Code Sight server. We'll call this my.codesight.box

10. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See *Platform specification for TeamForge 7.0* on page 324 for the full platform requirements.
- See *the SuSE Linux Enterprise Server deployment guide* for help.

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

11. Run the following command to install Black Duck Code Sight.

```
zypper install teamforge-codesearch
```

12. Copy the `site-options.conf` from the application box.
13. Run the installer.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

Do this on my.app.box:

14. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./bootstrap-data.sh
```

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.

Note:

- The TeamForge startup script installs the sample TeamForge project templates by default.
- If the project templates are already installed, you cannot re-install them using the TeamForge startup script.
- You may choose to delete the sample project templates. After deleting the sample project templates, you must set the *INSTALL_TEMPLATES* site options token to false. Otherwise, the project templates, if not found in the database, are installed automatically every time you restart the CollabNet services.

16. If you have installed Black Duck Code Sight, then install the license for Black Duck Code Sight. For more information, see *Install the Black Duck Code Sight license* on page 274.
17. If yo have installed Gerrit integration, integrate gerrit by running the `post-install.py` script.

```
/opt/collabnet/gerrit/scripts/post-install.py
```

18. Integrate the CLI reports by running the `post-install.py` script.

```
sudo /opt/collabnet/teamforge/runtime/scripts/post-install.py
```

19. Run the following initial ETL load jobs.
    a) Change to the `runtime/scripts` directory.

    ```
    cd /opt/collabnet/teamforge/runtime/scripts
    ```

    b) Run the `TrackerInitialJob`.

    ```
    ./etl-client.py -r TrackerInitialJob
    ```

    c) Run the `SCMInitialJob`.

    ```
    ./etl-client.py -r SCMCommitInitialJob
    ```

    Tip: For more information see *When do I run the initial load job?* on page 294.

20. Integrate Black Duck Code Sight with TeamForge by running the Black Duck Code Sight `post-install.sh` script.

    Note: It is assumed that Subversion's client configuration file (`/root/.subversion/config`) for the root user is the default one without customization.

    Note: If the Black Duck Code Sight is running on a separate server, run the following command in the code sight server.

    ```
    sudo /opt/collabnet/teamforge/runtime/scripts/codesearch/post-install.sh
    ```

21. After you run the Black Duck Code Sight `post-install.sh` script, run the following script from the same directory:

    ```
    ./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
    ```

    Provide a repository base URL similar to "http://myint.box.net/svn/repos" where `myint.box` is the box with the SCM integration server.

    Also, if you add a new integration server at some point later, you'll need to run this script, (after creating new integration server), in the TeamForge `app` box.

    Important: Do this if you have a multi-box setup with a SCM integration server on a separate server.

22. If the token *VALIDATE_SSL_CERTS* is set to "true", you must run the codesearch runtime script *trust-cert.sh* in the application box.

    ```
    cd /opt/collabnet/teamforge/runtime/scripts/codesearch
    ./trust-cert.sh
    ```

23. Restart the `collabnet` services.

    ```
    /etc/init.d/collabnet restart
    ```

24. Apply some finishing touches and make sure everything is running smoothly.
    a) 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.

    b) Create a sample project.

    See *Create a TeamForge project*.

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

    See *Create a site-wide broadcast*.

25. Revoke the super user permissions of database and datamart users.

    ```
    /opt/collabnet/teamforge/runtime/scripts/revoke-superuser-permission.py
    ```

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Install the Black Duck Code Sight license

To set up Black Duck Code Sight for TeamForge 7.0, you need to install a license.

1. Get the MAC Address of your Black Duck Code Sight server.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
sudo ./license-util.sh --listmacid
```

2. To receive the license, contact your CollabNet account manager or send an email request to info@collab.net.

   Note: A commercial TeamForge license is required; the TeamForge "free option license" does not qualify.

   a) Send the MAC Address of your Black Duck Code Sight server.
   You will receive the license file (in XML format).
   b) Preserve the license file for future use.

3. To install the license for Black Duck Code Sight, run these commands:

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
sudo ./license-util.sh --install <license_file_path>
```

   Type "Y" when prompted.

Bootstrap the Black Duck Code Sight instance

Follow these instructions to bootstrap the Black Duck Code Sight instance.

To bootstrap the Black Duck Code Sight instance:

1. Run the script in the command prompt.

```
cd $RUNTIME_DIR/scripts/codesearch
./bootstrap.sh
```

   Note: Type "Y" to continue with the bootstrap process.

2. Run the TeamForge installer.

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

3. Start the Black Duck Code Sight service.

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

If the Black Duck Code Sight is running on a separate server, then run the TeamForge installer in the Black Duck Code Sight box.

Upgrade Black Duck Code Sight on the same server

Follow these instructions to upgrade Black Duck Code Sight on the same server.

While it is possible to run Black Duck Code Sight on the same server as TeamForge, the best practice is to have Black Duck on a separate server.

1. Upgrade Red Hat Enterprise Linux / CentOS 6.3/6.4 and log in as root.

   - The host must be registered with the Red Hat Network. See *Platform specification for TeamForge 7.0* on page 324 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 7 for details.

3. Set the TeamForge repository configuration.

   If you are installing TeamForge with internet access:

- Contact the *CollabNet Support* and download the TeamForge 7.0 installation repository.
- Copy it to `/etc/yum.repos.d/`.
- Refresh your repository cache.

```
yum clean all
```

If you are installing TeamForge without internet access:

a) Contact the *CollabNet Support* to get the auxiliary installer package for TeamForge 7.0 disconnected installation and save it in `/tmp`.

OS	Installer package
Red Hat Enterprise Linux 6.3/6.4 32-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.i386.rpm`
Red Hat Enterprise Linux 6.3/6.4 64-bit	`CTF-Disconnected-media-7.0.0.1-380.el6.x86_64.rpm`

b) Unpack the disconnected installation package.

```
rpm -ivh <package-name>
```

c) **Note:** If the Red Hat installation DVD is mounted already, skip the following instructions. If not, mount the DVD.

Insert the Red Hat installation DVD.

This will provide some of the necessary software and utilities for installing TeamForge.

d) Mount the Red Hat installation DVD.

```
cd /media/
mkdir cdrom
mount /dev/cdrom ./cdrom/
```

Replace `cdrom` with the identifier for your server's CD drive, if necessary.

Tip: If there are any spaces in the automount, unmount it first, and then mount it as a filepath, without any spaces.

e) Create a yum configuration file (if it is not available) that points to the Red Hat installation DVD contents.

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

f) Copy this into the yum configuration file:

```
[RHEL-CDROM]
name=RHEL CDRom
baseurl=file:///media/cdrom/Server/
gpgfile=file:///media/cdrom/RPM-GPG-KEY-redhat-release
enabled=1
gpgcheck=0
```

g) Run the following commands to check if your yum configuration files are correct.

```
yum list httpd
yum list apr
```

Now the package manager will look for the Red Hat installation components on the DVD and not on the internet.

4. Install Black Duck Code Sight.

```
yum install teamforge-codesearch
```

5. Copy the master `site-options.conf` file from the `app` server and modify these tokens:

```
HOST_my.host.name=codesearch
DOMAIN_my.host.name=<mycodesearchbox.domain.com>
```

Note: If you have Black Duck Code Sight on SSL and the following external certificate tokens are not provided, the installer will generate a self-signed certificate.

```
For valid SSL certificates, configure the following tokens:
BDCS_SSL_CERT_FILE=
BDCS_SSL_KEY_FILE=
The ca.crt and chain files are optional -- leave out the tokens if you
 don't use the files.
BDCS_SSL_CA_CERT_FILE=
BDCS_SSL_CHAIN_FILE=
```

Advanced Black Duck Code Sight configuration settings:

Note: The following are the advanced configuration tokens which can be set once during the time of installation.

```
The path where the repositories enabled for codesearch to check out.
BDCS_SCAN_SOURCE_DIR_ROOT=/opt/collabnet/blackduck/scan

The path where the codesearch software is installed.
BDCS_INSTALL_PATH=/opt/collabnet/blackduck

The path where codesearch database is installed.
BDCS_PGSQL_HOME_DIR_ROOT=/opt/collabnet/blackduck/postgres

The port number for codesearch db server.
BDCS_PGSQL_PORT=55435

The tomcat maximum heap memory size in megabytes.
BDCS_TOMCAT_MX_IN_MB=1024

The shutdown port number for codesearch tomcat server.
BDCS_TOMCAT_SHUTDOWN_PORT=9189
```

6. Review the variables you've changed, then save the `site-options.conf` file.
7. Run the installer.

```
cd /opt/collabnet/teamforge-installer/7.0.0.1
./install.sh -r -I -V
```

8. To start the Black Duck Code Sight service, use:

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

To install the license for Black Duck Code Sight, follow *these instructions*.

Note: To migrate TeamForge repositories follow *these instructions*.

Set up Review Board

Review Board is a popular code review tool available with TeamForge 7.0 as a fully-integrated add-on.

Install Review Board

You must install Review Board before you can make it available as an integrated application to project managers on your TeamForge site.

To install Review Board successfully, ensure that other repositories similar to EPEL (Extra Packages for Enterprise Linux) are disabled apart from the CollabNet and Operating System repositories.

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

1. Download the `RBInstaller-1.3.0.0.22.zip` file from *http://collab.net/downloads/integrations#tab-1* and save it in the `/var/ops/` folder.

2. Unzip the `RBInstaller.zip` file.

```
cd /var/ops/
unzip RBInstaller-1.3.0.0.22.zip
```

3. Modify these values in the *install.conf* file to suit your installation environment.

```
sudo vi /var/ops/RBInstaller-1.3.0.0.22/installer/install.conf
```

Option	Description
rb_dir=/u1/reviewboard	The path of the directory where the Review Board files and libraries are installed.
rb_data_dir=/opt/collabnet/reviewboard/data	The path of the directory where Review Board's database file, review request files and attachments are stored.
domain=<domain name or host name>	The Review Board site information. For example, cu064.cloud.maa.collab.net.
rb_database_password=<reviewboard_db_password>	The Review Board database password.
rb_database_type=postgresql	The Review Board database type.
rb_database_name=<reviewboard_db_name>	The database name of the Review Board.
rb_database_user=<reviewboard_username>	The Review Board database user name.
rb_database_host=<reviewboard_db_hostname>	The Review Board database host name.
rb_database_port=<reviewboard_db_port>	The Review Board database port.
ctf_base_url=https://myapp.collab.net	The absolute URL of the TeamForge site that you want to associate to.
ctf_site_var_dir=/opt/collabnet/teamforge/var	The location of the rbctfevents.jar file.

4. Start the TeamForge application.

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

5. Before installing Review Board, you must know the password for the scmviewer account. Run the following commands to get the password:
   a) Run the grep command to get the encrypted password.

```
grep SCM_USER_ENCRYPTED_PASSWORD /opt/collabnet/teamforge/runtime/conf/
runtime-options.conf
```

   b) Use the *password_util.sh* script to decrypt the scmviewer password.

```
sudo /opt/collabnet/teamforge/runtime/scripts/password_util.sh -decrypt
'<value of SCM_USER_ENCRYPTED_PASSWORD>'
```

6. Run the following grep commands to get the value of HTTPD_USER, HTTPD_GROUP and HOME_DIR_BASE.

```
grep HTTPD_USER= /opt/collabnet/teamforge/runtime/conf/runtime-
options.conf
grep HTTPD_GROUP= /opt/collabnet/teamforge/runtime/conf/runtime-
options.conf
grep HOME_DIR_BASE= /opt/collabnet/teamforge/runtime/conf/runtime-
options.conf
sudo chown -R <HTTPD_USER>:<HTTPD_GROUP> <HOME_DIR_BASE>/<HTTPD_USER>
```

7. Run the installer in the Review Board installer directory.

```
cd /var/ops/RBInstaller-1.3.0.0.22
sudo python ./install.py -i -r --auth-scmuser
```

If this is an "advanced" mode installation (with the database on the same box or another box), do this on the database server.

8. Edit the `pg_hba.conf` as a postgres user and add the following entry at end of the file:

```
$ su - postgres
$ vi /var/lib/pgsql/9.0/data/pg_hba.conf
host ctfrbdb ctfrbuser <IP address of my.app.box>/32 md5
$ exit
```

9. Restart PostgreSQL.

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

10. Create the Review Board database and username.

```
$ su - postgres
$ createuser -P -S --no-createrole ctfrbuser
$ createdb -E UTF8 -O ctfrbuser ctfrbdb
$ exit
```

11. Restart PostgreSQL.

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

12. Set up the initial Review Board data. Run the script in the Review Board installer directory.

```
cd /var/ops/RBInstaller-1.3.0.0.22
sudo python ./bootstrap-data.py
```

13. Run the grep command to get the value of httpd_user.

```
grep httpd_user /etc/reviewboard.properties
sudo chown -R <httpd_user>:<httpd_user> /opt/collabnet/reviewboard/data
```

14. Restart the TeamForge application.

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

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

You should now have a Review Board instance ready to work with TeamForge. The installer has created two configuration files: `installer/conf/rb-application.xml` and `installer/conf/rb-deploy.xml`. See *Integrate Review Board with your TeamForge site* on page 285 for more information.

15. If SCM is installed on a separate box, run the following script to authenticate a scmviewer user against a TeamForge Subversion repository for creating a new review request.

```
sudo python ./svn-auth.py --repo-path=https://<scm_domain>/svn/repos/
<repo_dir_name>
```

Note: Run the following scripts available in the Review Board installer directory.

```
cd /var/ops/RBInstaller-1.3.0.0.22
```

- To recreate the runtime, run the `sudo python ./install.py -r` command.
- To update the revised password for a scmviewer account, run the `sudo python ./install.py --auth-scmuser` command.
- To change the scmviewer password, see *SCM_USER_ENCRYPTED_PASSWORD* on page 383.
- To uninstall Review Board, run the `sudo python ./install.py -u` command.

Install Review Board on TeamForge sites that use Oracle database

Use these instructions to install Review Board on TeamForge sites that use Oracle as its database. Note that the Review Board needs PostgreSQL database.

To install Review Board successfully, ensure that other repositories similar to EPEL (Extra Packages for Enterprise Linux) are disabled in addition to the Collabnet and Operating System repositories.

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

1. Install PostgreSQL.

   ```
   sudo yum install postgresql90-server postgresql90-libs
   ```

2. Download the `RBInstaller-1.3.0.0.22.zip` file from *http://collab.net/downloads/integrations#tab-1* and save it in the `/var/ops/` folder.
3. Unzip the `RBInstaller.zip` file.

   ```
   cd /var/ops/
   ```

   ```
   unzip RBInstaller-1.3.0.0.22.zip
   ```

4. Modify these values in the *`install.conf`* file to suit your installation environment.

   ```
   sudo vi /var/ops/RBInstaller-1.3.0.0.22/installer/install.conf
   ```

Option	Description
rb_dir=/u1/reviewboard	The path of the directory where the Review Board files and libraries are installed.
rb_data_dir=/opt/collabnet/reviewboard/data	The path of the directory where Review Board's database file, review request files and attachments are stored.
domain=<domain name or host name>	The Review Board site information. For example, cu064.cloud.maa.collab.net.
rb_database_type=postgresql	The Review Board database type.
rb_database_name=<reviewboard_db_name>	The database name of the Review Board.
rb_database_user=<reviewboard_username>	The Review Board database user name.
rb_database_password=<reviewboard_db_password>	The Review Board database password.
rb_database_host=<reviewboard_db_hostname>	The Review Board database host name.
rb_database_port=<reviewboard_db_port>	The Review Board database port.
ctf_base_url=https://myapp.collab.net	The absolute URL of the TeamForge site that you want to associate to.
ctf_site_var_dir=/opt/collabnet/teamforge/var	The location of the rbctfevents.jar file.

5. Start the TeamForge application.

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

6. Before installing Review Board, you must know the password for the scmviewer account. Run the following commands to get the password:
   a) Run the grep command to get the encrypted password.

      ```
      grep SCM_USER_ENCRYPTED_PASSWORD /opt/collabnet/teamforge/runtime/conf/
      runtime-options.conf
      ```

   b) Use the *password_util.sh* script to decrypt the scmviewer password.

      ```
      sudo /opt/collabnet/teamforge/runtime/scripts/password_util.sh -decrypt
      '<value of SCM_USER_ENCRYPTED_PASSWORD>'
      ```

7. Run the following grep commands to get the value of HTTPD_USER, HTTPD_GROUP and HOME_DIR_BASE.

   ```
   grep HTTPD_USER= /opt/collabnet/teamforge/runtime/conf/runtime-
   options.conf
   grep HTTPD_GROUP= /opt/collabnet/teamforge/runtime/conf/runtime-
   options.conf
   ```

```
grep HOME_DIR_BASE= /opt/collabnet/teamforge/runtime/conf/runtime-
options.conf
sudo chown -R <HTTPD_USER>:<HTTPD_GROUP> <HOME_DIR_BASE>/<HTTPD_USER>
```

8. Run the installer in the Review Board installer directory.

```
cd /var/ops/RBInstaller-1.3.0.0.22
sudo python ./install.py -i -r --auth-scmuser
```

9. Initialize the PostgreSQL database.

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

10. Edit the `postgresql.conf` file as a postgres user and update the following entry.

```
vi /var/lib/pgsql/9.0/data/postgresql.conf
listen_addresses = '127.0.0.1,<IP address of my.app.box>'
```

11. Edit the `pg_hba.conf` file as a postgres user and add the following entry at end of the file.

```
$ vi /var/lib/pgsql/9.0/data/pg_hba.conf
host ctfrbdb ctfrbuser <IP address of my.app.box>/32 md5
$ exit
```

12. Start the PostgreSQL service.

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

13. Create the Review Board database and user name.

```
$ su - postgres
createuser -P -S --createdb --no-createrole ctfrbuser
createdb -E UTF8 -O ctfrbuser ctfrbdb
$ exit
```

14. Restart PostgreSQL.

```
$ sudo /etc/init.d/postgresql-9.0 restart
```

15. Set up the initial Review Board data. Run the `bootstrap-data.py` script from the Review Board installer directory.

```
cd /var/ops/RBInstaller-1.3.0.0.22
sudo python ./bootstrap-data.py
```

16. Run the grep command to get the value of httpd_user:

```
grep httpd_user /etc/reviewboard.properties
sudo chown -R <httpd_user>:<httpd_user> /opt/collabnet/reviewboard/data
```

17. Restart the TeamForge application.

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

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

You should now have a Review Board instance ready to work with TeamForge. The installer has created two configuration files: `installer/conf/rb-application.xml` and `installer/conf/rb-deploy.xml`. See *Integrate Review Board with your TeamForge site* on page 285 for more information.

Note: Run the below scripts under the Review Board installer directory.

`cd /var/ops/RBInstaller-1.3.0.0.22`

To recreate the runtime, run the `sudo python ./install.py -r` command.

To update the revised password for a scmviewer account, run the `sudo python ./install.py --auth-scmuser` command.

To uninstall Review Board run the `sudo python ./install.py -u` command.

Note: To change the scmviewer password, see *SCM_USER_ENCRYPTED_PASSWORD* on page 383

Upgrade Review Board to latest build from CollabNet

In this task, we upgrade Review Board to its latest build.

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

1. Uninstall the existing Review Board application if you you are upgrading on the same box.

```
cd /var/ops/RBInstaller-1.2.0.0.138
sudo python ./install.py -u
```

2. Back up your Review Board data if you are upgrading TeamForge and Review Board on a separate server.
   a) The Review Board database backup is restored already when you have upgraded TeamForge. So, it is not necessary to take a backup of the Review Board database again.
   b) Backup the Review Board data directory.

```
cd /opt/collabnet
tar -zcvf /tmp/reviewboard_data.tgz reviewboard
```

   c) Copy `/tmp/reviewboard_data.tgz` to the `/tmp/ directory` of the new server.
3. Run the following command to update the `mod_python` from the Collabnet yum repo.

```
yum install mod_python
```

4. Unzip the `RBInstaller-1.3.0.0.22.zip`.

```
cd /var/ops/
sudo unzip RBInstaller-1.3.0.0.22.zip
```

5. Modify these values in the *`install.conf`* file to suit your installation environment.

```
sudo vi /var/ops/RBInstaller-1.3.0.0.22/installer/install.conf
```

Option	Description
rb_dir=/u1/reviewboard	The location of the Review Board installation directory where the Review Board files and libraries needed by Review Board is installed.
rb_data_dir=/opt/collabnet/reviewboard/data	The path of the ReviewBoard data directory where Review Board's database file, review request files and attachments will be stored.
domain= <domain name or host name>	The Review Board site information. For example, cu064.cloud.maa.collab.net
rb_database_password=<reviewboard_db_password>	The database password of Review Board.
rb_database_type=postgresql	The database type of Review Board.
rb_database_name=ctfrbdb	The database name of Review Board.
rb_database_user=ctfrbuser	The database username of Review Board.
rb_database_host=<database host name>	The database host name of Review Board. For example, cu064.cloud.maa.collab.net
rb_database_port=5432	The database port of Review Board.
ctf_base_url=https://cu064.cloud.maa.collab.net	The absolute URL of the TeamForge site that you want to associate to.
ctf_site_var_dir=/opt/collabnet/teamforge/var	The location of the rbctfevents.jar file.

6. Start the TeamForge application.

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

7. Before installing Review Board, you must know the password for the scmviewer account. Run the below commands to get the password:
   a) Run the grep command to get the encrypted password.

      ```
      grep SCM_USER_ENCRYPTED_PASSWORD /opt/collabnet/teamforge/runtime/conf/
      runtime-options.conf
      ```

   b) You will be provided with a password after running the script.

      ```
      sudo /opt/collabnet/teamforge/runtime/scripts/password_util.sh -decrypt
      '<value of SCM_USER_ENCRYPTED_PASSWORD>'
      ```

8. Run the following grep commands to get the value of HTTPD_USER, HTTPD_GROUP and HOME_DIR_BASE.

   ```
   grep HTTPD_USER= /opt/collabnet/teamforge/runtime/conf/runtime-
   options.conf
   grep HTTPD_GROUP= /opt/collabnet/teamforge/runtime/conf/runtime-
   options.conf
   grep HOME_DIR_BASE= /opt/collabnet/teamforge/runtime/conf/runtime-
   options.conf
   sudo chown -R <HTTPD_USER>:<HTTPD_GROUP> <HOME_DIR_BASE>/<HTTPD_USER>
   ```

9. Restore the Reviewboard data if you are upgrading TeamForge and Review Board on a new server.

   Note: Ensure that you have already copied the backup of the Review Board data directory in the `/tmp` directory of the new server from the old server.

   a) Restore the Review Board data directory.

      ```
      cd /opt/collabnet
      tar -zxvf /tmp/reviewboard_data.tgz
      ```

10. Run the installer in the Review Board installer directory.

    ```
    cd /var/ops/RBInstaller-1.3.0.0.22
    sudo python ./install.py -i -r --auth-scmuser
    ```

11. Run the following script in the Review Board installation directory.

    ```
    cd /var/ops/RBInstaller-1.3.0.0.22
    sudo python ./migrate.py
    ```

    Important: Ensure to run the migrate.py script, if you are reinstalling Review Board.

12. Restart the TeamForge application.

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

Upgrade Review Board-PostgreSQL on TeamForge sites that use Oracle database

Use these instructions to upgrade Review Board to its latest build on TeamForge sites that use Oracle as its database. Note that the Review Board needs PostgreSQL database.

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

1. Uninstall the existing Review Board application if you you are upgrading on the same server.

   ```
   cd /var/ops/RBInstaller-1.2.0.0.138
   sudo python ./install.py -u
   ```

Back up your Review Board data directory and database

2. Back up your Review Board data directory and database if you are upgrading TeamForge and Review Board on a new server.
   a) Back up the Review Board data directory.

      ```
      cd /opt/collabnet
      ```

```
tar -zcvf /tmp/reviewboard_data.tgz reviewboard
```

b) Back up the Review Board database.

```
su - postgres
pg_dump -C -p <database_port> ctfrbdb -f /tmp/rbdb.dmp
exit
```

c) Copy the `/tmp/reviewboard_data.tgz` and `/tmp/rbdb.dmp` files that you backed up to the `/tmp` directory of the new server.

3. On the new Review Board server, run the following command to update the `mod_python` from the CollabNet yum repo.

```
sudo yum install mod_python
```

4. Download the `RBInstaller-1.3.0.0.22.zip` file from *http://collab.net/downloads/integrations#tab-1* and save it in the `/var/ops/` folder.
5. Unzip the `RBInstaller-1.3.0.0.22.zip`.

```
cd /var/ops/
sudo unzip RBInstaller-1.3.0.0.22.zip
```

6. Modify these values in the *`install.conf`* file to suit your installation environment.

```
sudo vi /var/ops/RBInstaller-1.3.0.0.22/installer/install.conf
```

Option	Description
rb_dir=/u1/reviewboard	The path of the directory where the Review Board files and libraries are installed.
rb_data_dir=/opt/collabnet/reviewboard/data	The path of the directory where Review Board's database file, review request files and attachments are stored.
domain= <domain name or host name>	Review Board site information. For example, cu064.cloud.maa.collab.net.
rb_database_password=<reviewboard_db_password>	The Review Board database password.
rb_database_type=postgresql	The Review Board database type.
rb_database_name=ctfrbdb	The database name of the Review Board.
rb_database_user=ctfrbuser	The Review Board database user name.
rb_database_host=<database host name>	The Review Board database host name.
rb_database_port=<reviewboard_db_port>	The Review Board database port.
ctf_base_url=https://myapp.collab.net	The absolute URL of the TeamForge site that you want to associate to.
ctf_site_var_dir=/opt/collabnet/teamforge/var	The location of the rbctfevents.jar file.

7. Start the TeamForge application.

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

8. Before installing Review Board, you must know the password for the scmviewer account. Run the below commands to get the password:

a) Run the grep command to get the encrypted password.

```
grep SCM_USER_ENCRYPTED_PASSWORD /opt/collabnet/teamforge/runtime/conf/
runtime-options.conf
```

b) Use the *password_util.sh* script to decrypt the scmviewer password.

```
sudo /opt/collabnet/teamforge/runtime/scripts/password_util.sh -decrypt
'<value of SCM_USER_ENCRYPTED_PASSWORD>'
```

9. Run the following grep commands to get the value of HTTPD_USER, HTTPD_GROUP and HOME_DIR_BASE.

```
grep HTTPD_USER= /opt/collabnet/teamforge/runtime/conf/runtime-
options.conf
grep HTTPD_GROUP= /opt/collabnet/teamforge/runtime/conf/runtime-
options.conf
grep HOME_DIR_BASE= /opt/collabnet/teamforge/runtime/conf/runtime-
options.conf
sudo chown -R <HTTPD_USER>:<HTTPD_GROUP> <HOME_DIR_BASE>/<HTTPD_USER>
```

10. Do this if you are upgrading TeamForge and Review Board on a new server.
    a) Configure the database access. Edit the `pg_hba.conf` as a postgres user and add the following entry at end of the file:

```
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 app_box>'
```

    b) Start the PostgreSQL service.

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

    c) Create the Review Board database and user name.

```
su - postgres
createuser -P -S --createdb --no-createrole ctfrbuser
createdb -E UTF8 -O ctfrbuser ctfrbdb
exit
```

    d) Restart PostgreSQL.

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

11. Restore the Review Board data and database if you are upgrading TeamForge and Review Board on a new server.

Note: Ensure that the old Review Board data directory and database you backed up earlier has been copied to the `/tmp` directory of the new Review Board server.

    a) Restore the Review Board data directory.

```
cd /opt/collabnet
tar -zxvf /tmp/reviewboard_data.tgz
```

    b) Restore the Review Board database.

```
su - postgres
psql ctfrbdb < /tmp/rbdb.dmp
exit
```

12. Run the installer in the Review Board installation directory.

```
cd /var/ops/RBInstaller-1.3.0.0.22
sudo python ./install.py -i -r --auth-scmuser
```

13. Run the following script in the Review Board installation directory.

```
cd /var/ops/RBInstaller-1.3.0.0.22
sudo python ./migrate.py
```

Important: Run the `migrate.py` script, if you are reinstalling Review Board.

14. Restart the TeamForge application.

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

Integrate Review Board with your TeamForge site

When the Review Board application has been installed on your site, you can make it available for projects on your TeamForge site.

Review Board must be installed and configured before you can integrate it with your TeamForge site. See *Install Review Board* on page 276.

Important: You must not create, edit or delete new user accounts while installing the Review Board application and integrating it with TeamForge.

When you have installed Review Board, projects on your site can add Review Board to their set of collaboration tools.

1. Log into TeamForge as an administrator.
2. Click **Integrated Apps** in the Site Administration toolbar.
3. Click **Create**.
4. Use the **Browse** window to select the application and deployment configuration files that enable the Review Board application to work as a part of TeamForge:
   - `rb-application.xml` (Application configuration file): Contains the text strings for the Review Board user interface.
   - `rb-deploy.xml` (Deployment configuration file): Contains the data that Review Board 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 Review Board application is now available for all projects on your site. Project administrators can refer to the topic, *Add Review Board to a TeamForge project*, for instructions on adding Review Board 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.

TeamForge - Review Board integration: Install FAQ

Questions about TeamForge - Review Board integration.

What are the software requirements for installing Review Board as an integrated application in TeamForge 7.0?	Review Board can run on: • RHEL 6.3/6.4 • CentOS 6.3/6.4 • SUSE Linux 11 SP2 In addition, Review Board needs PostgreSQL 9.0.13. See *Software requirements for CollabNet TeamForge 7.0* on page 324 topic for more information.
Can I install TeamForge and Review Board on separate boxes?	No. Review Board must be installed on the same server where TeamForge runs.

TeamForge Review Board integration: General usage FAQ

General usage questions about the TeamForge - Review Board integration.

Which version of Review Board does TeamForge 7.0 support?	TeamForge 7.0 supports Review Board 1.6.3.
Which repositories does Review Board support?	Review Board supports only Subversion repository in TeamForge 7.0.
How do I manage users in Review Board?	You can manage Review Board users from TeamForge. Whenever you create or edit users in TeamForge, they are synchronized automatically in Review Board.
Can I specify 'RB' as a prefix in my project?	No. You cannot specify 'RB' as a prefix in your project. The prefix for Review Board must be unique for every project.
Is it possible to grant TeamForge specific-permissions as part of the system generated Review Board administrator?	No. It is not possible to grant TeamForge specific-permissions as part of the system generated Review Board administrator (integrated application specific role).
Can I use the Review Board 'Search' feature after integrating Review Board with TeamForge?	No. TeamForge does not support the 'Search' feature of Review Board.
What are the additional features available in Review Board after you integrate it with TeamForge?	Review Board uses some of the TeamForge features like object IDs, links, GO URLs, and SVN integration and associations. For more information, see *How does an integrated application interact with other TeamForge tools?* on page 316
What are the other TeamForge features which Review Board does not support after you integrate Review Board with TeamForge?	Global search, page component, recent history and project template features of TeamForge are not supported in Review Board.
Where can I find the documentation for Review Board?	You can find the documentation for Review Board *here*.

Set up the TeamForge Git integration

The TeamForge Git integration is supported on RedHat and Centos version 5.6 or later. With TeamForge 7.0 (and later), you can install the Git integration is a part of installing TeamForge.

You can install Git integration on a separate box and other services on the main applicaton box. For more information, see *Install TeamForge 7.0 with Git integration on a separate box* on page 28

You can upgrade to TeamForge 7.0 with the Git integration on the same server, or with TeamForge and the Git integration on separate servers. See *Upgrade the Git integration with TeamForge.*

After you've set up the TeamForge Git integration, it is possible to modify the settings. See *Reconfigure TeamForge Git integration.*

To uninstall the integration, use the YUM utility. For more information see *Uninstall the TeamForge Git integration.*

For more information on the functionality and features provided by the Git integration for TeamForge 7.0, see *TeamForge Git integration.*

A preface to the TeamForge-Orchestrate integration

Though the procedure to add the TeamForge Orchestrate, as an integrated application, is similar to adding other integrated applications, consider the following while integrating TeamForge and Orchestrate.

Uploading the event handler JAR file

After adding the *TeamForge Orchestrate* to your site, it is mandatory to upload the custom event handler JAR file to get the pre and post commit notifications. *Click here* for steps to add a custom event handler.

Implications of TeamForge-Orchestrate integration on SCM commits and associations

Pre and post commit notifications will be sent from TeamForge to Orchestrate only if the "***Association Required on Commit***" is enabled for the repository and if the "***require-scm-integration***" is set to "true" for the Orchestrate application.

Post TeamForge-Orchestrate integration, SCM commits will fail in TeamForge if all the following conditions are true and if the Orchestrate application is down or if there are errors while processing the commit request (on the Orchestrate's side):

- The **Association Required on Commit** is enabled for a repository.
- The **require-scm-integration** is set to "true".
- The commit message contains the integrated application's ID. In other words, the Orchestrate application's ID in this case.

SCM commits can also fail if the custom event handler exists in the TeamForge even after the removal of the Orchestrate application from the TeamForge. It is highly recommended to remove the custom event handler once you remove the Orchestrate application.

Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see *TeamForge Orchestrate installation.*

Frequently asked questions about administrating a TeamForge site

Use this background knowledge to help you install, maintain and support a CollabNet TeamForge site.

TeamForge installation FAQs

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 *Platform specification for TeamForge 7.0* on page 324.

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 246 for information.
- If your site requires SELinux, you must configure your Apache service. See *Set up SELINUX* on page 245 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.

Should I upgrade to TeamForge 7.0 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.

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.

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.

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.

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.<site>/sf/global/jsp/
buildtime.html"
    format="html" scope="external">here</a>.</p>
<hr> <address>Apache/2.2.3 (Red Hat) Server at www.<site>.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 does the the Yum installer display a warning message on Centos 6?

The yum installer displays a warning message on Centos 6 if you are not using the latest version update of Centos or if the new GNU Privacy Guard (GPG) keys are not in place.

On the Centos system, the new GPG keys are generally sent as an update before the expiry of the old keys. You must perform the version update regularly to receive the new GPG keys.

To avoid the warning message, use the "-nopgcheck option" in the yum install command.

```
yum install --nogpgcheck teamforge
```

Why am I getting a Yum repository filename conflict?

This means that you have created a repository filename starting with `collabnet`.

The repository file for the TeamForge installer begins with "collabnet-*.repo". If you have a customized repository with the same name, then the TeamForge installer renames the repository file with ".backup" as an extension and

installs the repository file. It is better not to have any repository file name that begins with `collabnet` apart from the one that is installed by the TeamForge installer.

When do I run the initial load job?

You can run the initial load job any time after the site is upgraded to TeamForge 7.0. We recommend that you run it before you hand over the site to the users.

You can run the job when the site is in use. On an average, there might be a ten percent degradation in the response time when the job is running. If the job is not run as a part of the upgrade, we recommend that you run it at a time when the site usage is relatively low. The initial load job may consume more time to complete and this depends on the data available in the site. So this job should be triggered when the site is stable after the upgrade and the chances of site shut down is remote.

Note: The initial load job can be run immediately in case of a fresh installation.

How do I list all the ETL jobs?

You can list all the ETL jobs by running the following command.

```
./etl-client.py -a
```

How do I run the initial load jobs?

Run the scripts specified below to run the initial load jobs.

```
$ [RUNTIME_DIR]/scripts/etl-client.py -r SCMCommitInitialJob
```

```
$ [RUNTIME_DIR]/scripts/etl-client.py -r TrackerInitialJob
```

What happens if I run an initial load job that has already run successfully?

Re-running the initial load job does no harm. The system performs an internal check and the job is aborted.

How do I check the status of the job?

Site administrators will be notified through mails if there is a job failure. The following command will help you identify whether a job is currently running or not.

```
$ [RUNTIME_DIR]/scripts/etl-client.py -a
```

What happens if there is a job failure?

An email will be sent to site administrators in case of a failure. The job should be re-triggered manually. Data will be harvested from the last completed milestone prior to the failure.

Is it mandatory to run the initial load job for a new site?

Yes, it is mandatory to run the initial load job for a new site.

What happens if I miss running the initial load during the installation or upgrade?

You can run this any time after the site is upgraded to TeamForge 7.0. However, the incremental data harvesting is blocked until the initial load is run. This is true for an existing site that is upgraded to TeamForge 7.0 as well as for a new site.

Why does the incremental_etl job fail when the initial load job is not triggered?

The incremental load is made to fail if the initial load is not run. You can ignore this.

Common errors in TeamForge

Common problems and solutions in TeamForge.

Troubleshooting VMware errors

Common problems and solutions related to VMware.

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/teamforge/buy-it*

  .
- 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 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/*

Troubleshooting database/datamart/ETL errors

Common problems and solutions related to database/datamart/ETL.

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-ds.xml, located in `/opt/collabnet/teamforge/runtime/jboss/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 am I getting a 'Not running' message when the Datamart service is stopped?

When TeamForge and Datamart are running in a single instance, the TeamForge database is stopped when you stop the Postgres services. The message, 'Not running' is displayed when you stop the Datamart service. You can ignore this message.

Why am I getting an email specifying that the ETL job has failed?

You are getting this email because one of the Extract transformation and load (ETL) jobs has failed during the run. You can see the etl.log for more details to find out the reason for the job failure.

The ETL job failure may happen because of the following reasons:

- Out of memory error.
- No response from the database.

If the ETL job failure is happening for the first time, you can restart the ETL (`[RUNTIME_DIR]/scripts/collabnet restart etl`) and check if the problem is occurring again. You can increase the JVM heap size by specifying the same in ETL_JAVA_OPTS if the problem keeps recurring. The default value is `-Xms160m -Xmx256m`. You can increase the heap size depending on the memory available in the box.

Check if both TeamForge and Datamart are up and responding to queries if there is no response from the database. Restart the ETL (`[RUNTIME_DIR]/scripts/collabnet restart etl`).

Contact the Collabnet support if the problem persists.

Why am I not able to see the status of the Postgres in the collabnet startup script?

You may not be able to see the status of the Postgres if the host name of the *HOST_ token* is set to localhost in a SaaS multibox setup.

The Teamforge installer fails to add the IP address of the database box to the listen address in the `postgresql.conf` file if the host name of the *HOST_ token* is set to localhost in a SaaS multibox setup.

Note: You must add the IP address of the database box to the listen address in the `postgresql.conf` file.

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
```

Troubleshooting JBoss errors

Common problems and solutions related to JBoss.

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 is the site slow after upgrading from CollabNet TeamForge 6.2 to 7.0 Patch 1?

This is because of the stability issues with the JVM option -XX:+TieredCompilation.

The TieredCompilation option is enabled by default in the JBoss Java virtual machine. To improve the performance, it is mandatory to turn it off by adding -XX:-TieredCompilation to the existing list of options in the *JBOSS_JAVA_OPTS* token in the `site-options.conf` file.

Troubleshooting E-mail errors

Common problems and solutions related to E-mails.

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 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/teamforge/runtime/scripts/phoenix.sh stop
sh /opt/collabnet/teamforge/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:

```
4D61696C313137303833323131343031322.Repository.?FileObjectStore
4D61696C313137303833323131343031322.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 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.

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/runtime/james/apps/james/SAR-INF/config.xml`:

```
<gateway>smtp.example.com</gateway>
<gatewayPort>25</gatewayPort>
```

If your gateway mail server requires authentication to send email, you may also add the following directives:

```
<username>username</username>
<password>password</password>
```

Troubleshooting other errors

Common problems and solutions.

How do I enable the Postgresql log files archiving when the services are not started using the CollabNet startup script?

The Postgresql log files archiving can be enabled by running a simple command.

It is recommended you use the CollabNet init script to start and stop the Postgres services. However, if you use the Postgresql init script to start or stop the Postgres services, the postgresql log files are not archived by default.

To enable the Postgresql log files archiving, run the following command:

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

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.

Why am I not able to see the charts for tracker metrics?

You may not be able to see the charts for the tracker metrics if the tracker initial load is not running correctly.

The incremental data collection is disabled until the initial load is run. You can check if the initial load is completed successfully by executing the query below from the Ad-hoc reporting page against the datamart.

```
select status from etl_job where job_name='tracker_initial_etl';
```

You must get the status value as 1 if the initial load is completed successfully. Otherwise, you must trigger the job manually by executing the command: `[RUNTIME_DIR]/scripts/etl-client.py -r TrackerInitialJob`

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.

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 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/log/integration/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 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.

Why am I not getting any error messages when executing the Subversion upgrade script?

Error messages may come when Subversion is installed with a dependent package from an unknown source.

The Subversion working copy script assumes that Subversion is installed with the dependent packages from a proper source repository(RHEL/CollabNet). If you install any dependent packages from any unknown source that is not authorized by RHEL/CollabNet, it will result in inconsistency and this cannot be handled by the Subversion working copy script.

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%' ;
```

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 246 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.

Open the REPORTS_DATABASE_PORT if you are granting direct access to the datamart from specific IPs using the REPORTS_DB_ACCESS_HOSTS `site-options.conf` token.

Ports to be open in the firewall environment for TeamForge 7.0

Source Box	Service	Port	Open type
APP	Apache	80/443	Universal
APP	TeamForge Database	5432	TeamForge
APP	SCM Integration	7080	TeamForge
APP	Git Integration	9081	TeamForge
APP	Git listener	9080	TeamForge
APP	Git	29418	Universal
APP	Indexer	2099	TeamForge
APP	CVS	2401	Universal
REPORTS	Reports Database	5632	TeamForge

Source Box	Service	Port	Open type
REPORTS	Tomcat ETL	7010	TeamForge
CODESEARCH	Tomcat CS	9180	Universal

Note:

- APP-> Server that runs the TeamForge application, SCM integration and operational database services.
- REPORTS-> Server that runs the Datamart and ETL services.
- CODESEARCH-> Server that runs the Black Duck Code Sight services.
- Universal-> Open this port for global access.
- TeamForge-> Open this port only for TeamForge servers.

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.

Concepts and terms in TeamForge

Descriptions of concepts and terms in TeamForge.

Advantages of using the Apache TIKA parser library for indexing

Starting TeamForge 7.0, the underlying parser library for indexing has been changed from Stellent to Apache TIKA.

The Apache TIKA parser library has the following advantages over the Stellent parser library:

Issue	Stellent	Apache TIKA
Stale process issue	Parsing of corrupt or unrecognized files by the Stellent parser libraries often result in stale processes that consume swap space and add to the load on the system, which may at times lead to site outage. To manage such processes, you may choose to create and deploy stale process monitors and the stale processes, when detected, must be removed manually to prevent site outage.	Parsing of unrecognized or corrupt files by Apache TIKA libraries is robust and needs no manual intervention as there are no stale process issues.
Search queue processing speed	It takes five minutes to timeout when the Stellent parser library encounters a corrupt or unrecognized file that it knows not how to parse. If there are more such corrupt or unrecognized files, more time is wasted by the indexer waiting for a response (or a timeout) from the Stellent parser, which in turn adversely impacts the search queue processing speed.	The Apache TIKA parser library is capable of determining whether a file it encounters can be parsed or not. As no time is wasted by the indexer waiting for a response (or a timeout) from the parser, the search queue processing speed is better with the Apache TIKA.
Multiple processes Vs Single JVM	For parsing files, the Stellent parser library spawns one subprocess per file. Meaning, the number of subprocesses is equal to the number of files to be parsed and it is possible that we may end up with the stale process issue as discussed earlier. As a result, if the Stellent processes consume	The Apache TIKA, being a Java-based parser library, works within the JVM and makes the external resource pool available exclusively for other processes and applications. As the search JVM, where the Apache TIKA library lives, can also be separated starting TeamForge 7.0, it can be managed better.

Issue	Stellent	Apache TIKA
	more resources, other processes and applications are left with scarce resources.	

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.

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?

An ETL application extracts 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.

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.

Object IDs

Integrated application object IDs are of the form "`prefix_objectId`". 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.

In an object ID such as "`prefix_objectId`", the "`prefix`" is case-insensitive, whereas the "`objectId`" is case-sensitive. For example, the two object IDs, "`PT_SC1`" and "`pt_SC1`", refer to the same object in TeamForge. Whereas, the two object IDs, "`PT_SC1`" and "`PT_sc1`", refer to two different objects in TeamForge. Here, `PT` and `pt` are case-insensitive and the `SC1` and `sc1` are case-sensitive.

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.

Setting up multiple prefixes for integrated applications

At times, you may want to use more than one prefix for an integrated application such as the TeamForge Orchestrate. It is possible to have multiple prefixes set up for integrated applications. You must have the prefixes, separated by commas, included in the XML application configuration file and upload the file to your TeamForge site. For more information about uploading the application configuration file, see *Edit an integrated application*.

Consider the following while setting up multiple prefixes for an integrated application:

- Prefixes, once set up using the XML application configuration file, cannot be modified.
- A prefix can be up to six alpha-numeric characters in length. However, the combined length of all the prefixes cannot exceed 128 alpha-numeric characters.
- The "require-per-project-prefix" must be set to false in the application configuration file. In case it is set to true, an error message appears when you upload the application configuration file.
- Do not use existing prefixes. You cannot upload an application configuration file consisting of one or more prefixes already in use in TeamForge.

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.

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.

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.

Are role-based permissions allowed for sub-folders in the TeamForge Documents?

Yes. The TeamForge Administrator can give permission to access sub folders in the TeamForge Documents based on the user roles, using the Roles option.

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.

Tasks in TeamForge

Questions about performing specific tasks in TeamForge.

How do I change the time to run the ETL jobs?

The *ETL_JOB_TRIGGER_TIME* can be modified to specify a different time.

By default, the ETL job runs at 2:30 AM (local time) everyday. It is recommended to run this once daily to avoid any performance degradation of the Teamforge site. See *ETL_JOB_TRIGGER_TIME* on page 361 for more information.

How can I check the status of ETL?

The `[RUNTIME_DIR]/scripts/collabnet status etl` displays the status of the ETL process.

You can get additional information about the various ETL jobs that are configured using the command `[RUNTIME_DIR]/scripts/etl-client.py -a`

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 254 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 TeamForge log files* on page 256.

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 to reinstall a deleted installation directory?

You can reinstall an installation directory if it has been deleted inadvertently.

Before attempting to reinstall the deleted installation directory the remnants must be wiped out completely. Rather, remove the `collabnet-local.repo` file under `/ etc/yum.repos.d` before reinstallation. This is mandatory for a clean and complete installation.

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 Datamart?

You can use the *psql-reporting-wrapper* script to connect to the datamart.

Usage

Run this script as below:

```
sudo [RUNTIME_DIR]/scripts/psql-reporting-wrapper
```

How do I connect to the Teamforge Postgres database?

You can use the *psql-wrapper* script to connect to the TeamForge application database.

Usage

Run this script as below:

```
sudo [RUNTIME_DIR]/scripts/psql-wrapper
```

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

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.

Reference information about TeamForge

Use this reference information to get deeper detail on configuration files, logs, scripts and other resources you use to administrate TeamForge.

Platform specification for TeamForge 7.0

This is the hardware and software platform that TeamForge 7.0 runs on.

Hardware requirements for CollabNet TeamForge 7.0

The following hardware is recommended for the TeamForge application and database servers for sites having up to 100 users.

- 2 x CPU 2GHz
- 4 GB RAM (but 16 GB is good for large sites)
- 40 GB hard drive

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

Recommendations

It is highly recommended that you install the TeamForge application and database on separate 64 bit physical servers. Each server must meet the same dual-processor, 2-GHz standard.
While it is possible to run Black Duck Code Sight on the same server as TeamForge, the best practice recommended by Black Duck is to install on a separate 64 bit server. The 32 bit server is recommended for evaluation purpose or for personal use only.
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.
As a result of significant infrastructure and platform changes for TeamForge 7.0 (JBoss 7.0, Postgres 9.0.13, *Tika*, Red Hat Linux 6.3 and so on), it is highly recommended that you add more RAM, which is to be exclusively allocated to Postgres. CollabNet recommends adding 50% more RAM for 7.0 compared to TeamForge 6.2. Contact the CollabNet Support for tuning the application for the additional RAM.

Software requirements for CollabNet TeamForge 7.0

This is the official list of software tested for compatibility with CollabNet TeamForge 7.0.

Software version requirements for 32/64 bit platform

		Red Hat Enterprise Linux 6.3/6.4[#]	CentOS 6.3/6.4	SUSE Linux Enterprise Server 11 SP2[#]
Database	PostgreSQL	9.0.13	9.0.13	9.0.13
	Oracle DB[#]	11G (R1 and R2)	11G (R1 and R2)	11G (R1 and R2)
	Oracle Client[#]	11g	11g	11g
Java[#]	JDK	1.7.0_21	1.7.0_21	1.7.0_21

	Red Hat Enterprise Linux 6.3/6.4[#]	CentOS 6.3/6.4	SUSE Linux Enterprise Server 11 SP2[#]
Jboss	7.1.1	7.1.1	7.1.1
Tomcat	7.0.22	7.0.22	7.0.22
Open LDAP	2.4.23	2.4.23	2.4.26
Quartz Job Scheduler	1.8.3	1.8.3	1.8.3
Browser Google Chrome	24, 25 and 26	24, 25 and 26	24, 25 and 26
Mozilla Firefox	18, 19 and 20	18, 19 and 20	18, 19 and 20
Microsoft IE	8, 9 and 10	8, 9 and 10	8, 9 and 10
Package Managers	The `yum` package manager is supported up to and including version 3.2.29 The `zypper` package manager is supported up to and including version 1.6.161		
CollabNet TeamForge 7.0 supports the following software versions for integration			
Black Duck Code Sight[#]	2.1.3	2.1.3	2.1.3
Review Board[#]	1.6.3	1.6.3	1.6.3
GIT	1.7 or later	1.7 or later	1.7 or later
Subversion	1.7.11	1.7.11	1.7.11
Subversion Edge	3.3.1	3.3.1	3.3.1
CVS	1.11.x	1.11.x	1.11.x
Perforce	2012	2012	2012
ViewVC	1.1.20	1.1.20	1.1.20

- [#] Red Hat Enterprise Linux servers must have access to the *Red Hat Network* or equivalent (satellite server, spacewalk, or RHN proxy).
- [#] SUSE Linux Enterprise Server must be registered with Novell.
- [#] Oracle Express Edition is not supported.
- [#] For larger installs, the max JVM heap size should be increased to 1024MB (or larger), depending on available resources on the server.
- [#] Black Duck Code Sight 2.1.3 tested with PostgreSQL 9.0.13 only. Black Duck Code Sight with Oracle was not tested.
- [#] Review Board 1.6.3 tested with PostgreSQL 9.0.13 only. Review Board with Oracle was not tested.

CollabNet TeamForge 7.0 on VMware and ESXi

The CollabNet TeamForge 7.0 is tested for compatibility with VMware Player 5.0 or later. For Windows and Linux versions supported by VMware Player 5.0 or later, see *VMware Player Documentation*.
The CollabNet TeamForge 7.0 is tested for compatibility with ESXi 5.0 or later.

Microsoft applications

The following Microsoft applications have been tested with CollabNet TeamForge:

- Microsoft Project 2002 (with Service Pack 1) on Windows XP Service Pack 2 and Windows 2000 Service Pack 4.
- Microsoft Project 2003 (with Service Pack 1) on Windows XP Service Pack 2 and Windows 2000 Service Pack 4.
- Microsoft Office XP (with Service Pack 3) on Windows XP Service Pack 2 and Windows 2000 Service Pack 4.
- Microsoft Office 2003 (with Service Pack 1) on Windows XP Service Pack 2 and Windows 2000 Service Pack 4.

Versions of RPM packages in Red Hat and CentOS installations

Make sure that you have installed the correct versions of the required packages.

The following table lists the RPM package version requirements for Red Hat and CentOS installations.

Package	Version Number
createrepo	0.9.9-17 or later
cvs	1.11.x
cx_oracle	5.0.4-1
httpd (Red Hat)	2.2.15-15
httpd (CentOS)	2.2.15-28
jdk	1.7.0_21
mod_dav_svn	1.7.9-1
mod_python	3.3.1-15
mod_ssl (Red Hat)	2.2.15-15
mod_ssl (CentOS)	2.2.15-28
neon	0.29.3-2 or later
oracle-instantclient11.2-basic	11.2.0.1.0-1
python	python-fpconst 0.7.3-6.1 python-lxml - 2.2.3-1 python-chardet - 2.0.1-1 python-curl - 7.19.0-8
rcs	5.7-37
subversion	subversion - 1.7.9-1 subversion-python - 1.7.9-1 subversion-perl - 1.7.9-1 subversion-devel - 1.7.9-1
yum	yum - 3.2.29 yum-metadata-parser - 1.1.2-16 yum-security - 1.1.30 yum-utils - 1.1.30
ZSI	2.0-6 or later.

Versions of RPM packages in SUSE installations

Make sure that you have installed the correct versions of the required packages.

The following table lists the RPM package version requirements for SUSE installations.

Package	Version Number
apache2	2.2.12
apache2-mod_python	3.3.1-147
apache2-worker	2.2.12-1.28
libapr1	1.3.3-11
libapr-util1	1.3.4-12
libneon27	0.29.6-6
PyGreSQL	4.0-392
SOAPpy	0.11.6
createrepo	0.4.11-82
cvs-stable	1.11.23
cx_Oracle	5.0.4-1
jdk	1.7.0_21
neon	0.29.6-6
oracle-instantclient11.2-basic	11.2.0
python	python-fpconst - 0.7.2-3 python-lxml - 2.1.2-2.4 python-chardet - 1.0.1-1 python-curl - 7.19.0
subversion	subversion - 1.7.9-1 subversion-python - 1.7.9-1 subversion-perl - 1.7.9-1 subversion-devel - 1.7.9-1
ZSI	2.0-6 or later.

Scripts installed with TeamForge 7.0

System administrators can use these utilities to control the behavior of the application.

backup-rb-data.py

The `backup-rb-data.py` script is used to back up the Review Board application data.

Overview

The Review Board application data includes the database and files. If there are any files in the backup directory, the script overwrites these files.

Usage

Run this script as follows:

```
python ./backup-rb-data.py --backupdir={dir}
```

Options

The following options are available for the `backup-rb-data.py` script:

-b | --both	Back up both the database and the filesystem. This is the default option.
-d | --database	Back up the database.
-f | --files	Back up the filesystem.
-h | --help	Provides a list of all available options for this script.

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.

CodeSightMigration.sh

The `CodeSightMigration.sh` script is used to create Black Duck Code Sight projects for existing TeamForge repositories. After migration, the code search functionality is enabled for migrated repositories.

Location

`runtime/scripts/codesearch/CodeSightMigration.sh`

Usage

Before running this script, do the following:

- Turn on the CODESEARCH_ENABLED flag on the TeamForge server.
- Make sure the tokens specific to Black Duck Code Sight, such as the server name and port, are valid.

Note: Run this script on the TeamForge `app` box as the site admin user.

Example

The following command enables Code Search for the given repositories.

```
[root@cuXX codesearch]# ./codesightmigration.sh --filename=/tmp/repository-ids.txt
```

Targets

The following targets are available for this script:

--target=create	This is the default target and is used for creating Black Duck Code Sight projects for existing TeamForge repositories.	
	[--filename=<repositories-input-filename>]	Default: /tmp/repository-ids; specify any other filename with the --filename parameter.
	--username=<admin-username>	Site admin username
	--password=<admin-password>	Site admin password
	[--totalthreads=<number of threads to spawn>]	Default: 25; the total number of threads needed to run Black Duck Code Sight migration.
--target=delete	Delete Black Duck Code Sight projects for the corresponding TeamForge repositories.	
	[--filename=<repositories-input-filename>]	Default: /tmp/repository-ids; specify any other filename with the --filename parameter.
	--username=<admin-username>	Site admin username
	--password=<admin-password>	Site admin password

	[--repoids=<repo-id1,repo-id2,repo-id3>] Default: delete all repositories as specified in the input filename or delete the repositories specified in the command line
--target=status	Status of Black Duck Code Sight projects created for TeamForge repositories.
	[--filename=<repositories-input-filename>] Default: /tmp/repository-ids; specify any other filename with the --filename parameter.
	--username=<admin-username> Site admin username
	--password=<admin-password> Site admin password
	[--repoids=<repo-id1,repo-id2,repo-id3>] Default: status of all repositories as specified in the input filename or the status of the repositories specified in the command line

Logs

Three types of logs are available. They are located at `site/log/codesearch`.

- `codesightdebugmessage.log`: used for debugging Black Duck Code Sight issues
- `codesightprojectstatus.log`: shows project status for repositories; created when target = status is run
- `codesightfailedrepositories.log`: shows project status for failed project repositories; created when target = status is run

Comments

There is a another script that generates the final SQL to fetch repository ids for a given list of repository paths. See *repository-sql.sh* for more information.

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.py

The `domain_change_db.py` 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.py [--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.py` 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.
--threadlimit	Defines the maximum number of simultaneous threads that can be invoked by this program. The default value is '50'

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.
--threadlimit	Defines the maximum number of simultaneous threads that can be invoked by this program. The default value is '50'

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 the ETL service.
-a | --status-all	Prints the status of incremental and historical jobs configured in the ETL service.

-v| --verbose	Chronicles the process of requested operation a bit more.
-r| --run=	Triggers a job manually for a given job.

Data collection (ETL) jobs supported by the ETL service

While some ETL jobs are scheduled to run automatically, some must be triggered manually. The following table lists the available ETL jobs in TeamForge.

Job Category	Job Name	Description
Historical Data Collection Historical data collection jobs must be triggered manually. As a best practice, these jobs are run as part of post migration activities. Refer to the "Related Links" for more information.	SCMCommitInitialJob	Collects the historical commit data from TeamForge.
	TrackerInitialJob	Collects the historical data of artifacts from TeamForge.
Incremental Data Collection Incremental data collection jobs collect data that are added or modified incrementally on an ETL run-to-run basis. These jobs are scheduled to run automatically on a regular basis.	SCMCommitActivityJob	Collects the SCM commit data incrementally on an ETL run-to-run basis.
	TrackerIncrementalJob	Collects the tracker artifacts data incrementally on an ETL run-to-run basis.
	UserLoginActivityJob	Collects the user logon activity data incrementally on an ETL run-to-run basis.
Imported Data Collection (Simbel) TeamForge supports bulk data import through Simbel. This job collects Simbel-imported data. This job must be triggered manually post data import into TeamForge.	SimbelImportJob	Collects all Simbel-imported data such as the user logon activity, SCM commit and tracker artifacts data.

fix_data_permission.sh

The `fix_data_permission.sh` script is used to fix the TeamForge filesystem permissions when you backup and restore the TeamForge data in a new box during TeamForge upgrade.

Usage

Run this script as follows:

```
fix_data_permissons.sh [-r <repo_thread_count>>] [-d <dir_thread_count>] [-t
<fix_permission item>]
```

Example

Run the following command to fix the permission of GIT repositories.

```
./fix_data_permissons.sh -t git
```

Run the following command to fix the permission of GIT repositories and app directories.

```
./fix_data_permissons.sh -t git -t app
```

Run the following command to fix the permission of 15 Subversion repositories simultaneously.

```
./fix_data_permissons.sh -r 15 -t svn
```

Options

The following options are available for the `fix_data_permission.sh` script:

-r	Number of repos to process at a time. Default=5
-d	Number of directories to process at a time. Default=<No of processors>
-t	Item to fix. Possible values are [app, http, svn, cvs, git]. Default=<all items>
-h | --help	Prints the help information.

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 | --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.

password_util.sh

The `password_util.sh` script is used to get the encrypted or decrypted password value for the user `scmviewer`.

Usage

To encrypt:

```
sudo /opt/collabnet/teamforge/runtime/scripts/password_util.sh -encrypt
'teamforge'
```

```
[root@xx scripts]# ./password_util.sh -encrypt 'teamforge'
Input String:teamforge
Encrypted password:VBxJJvzbXb5tNx2SxR26egA==
```

To decrypt:

```
sudo /opt/collabnet/teamforge/runtime/scripts/password_util.sh -decrypt
'VBxJJvzbXb5tNx2SxR26egA=='
```

```
[root@xx scripts]# ./password_util.sh -decrypt 'VBxJJvzbXb5tNx2SxR26egA=='
Input String:VBxJJvzbXb5tNx2SxR26egA==
Decrypted password:teamforge
```

psql-wrapper

The `psql-wrapper` script is used to connect to the TeamForge application.

Usage

Run this script as below:

```
sudo [RUNTIME_DIR]/scripts/psql-wrapper
```

Comments

- Run this script as a sudo user.
- Run this script with the postgres backend.
- You have full write access to the database for executing queries.

Note: This script is not supported in the Oracle backend.

psql-reporting-wrapper

The `psql-reporting-wrapper` script is used to connect to the datamart.

Usage

Run this script as below:

```
sudo [RUNTIME_DIR]/scripts/psql-reporting-wrapper
```

Comments

- Run this script as a sudo user.
- Run this script with the postgres backend.
- You have full write access to the datamart for executing queries.

Note: This script is not supported in the Oracle backend.

restore-data.py

The `restore-data.py` script is used to restore the Review Board application data from the backup directory.

Overview

This script removes the existing Review Board application data present in the system and restores data from the backup directory.

Usage

Run this script as follows:

```
python ./restore-rb-data.py –backupdir={dir}
```

Options

The following options are available for the `restore-data.py` script:

-b | --both	Restore both the database and the filesystem. This is the default option.
-d | --database	Restore the database.
-f | --files	Restore the filesystem.
-h | --help	Provides a list of all available options for this script.

SearchReindex.py

The `SearchReindex.py` script allows you to reindex the entire TeamForge data.

Overview

You can use this script to reindex the entire TeamForge data or you can choose to reindex the subset of data types.

Usage

Run this script as follows:

```
./SearchReindex.py --<component name>
```

Example

To perform a search reindex for the tracker, run this command:

```
./SearchReindex.py --trackers-only
```

To perform a search reindex for the wiki, run this command:

```
./SearchReindex.py --wiki-only
```

To perform a search reindex for documents run this command:

```
./SearchReindex.py --documents-only
```

Options

--single-item itemId, | -i	Schedules a re-index for just the given item. If the item id is for a project the scheduling results in the server re-indexing all of the project data.
---force-index | -f	Force indexing (doesn't check if item is searchable already).
--artifacts only | -a	Reindex all artifacts on the site that are currently not searchable or all artifacts if option f is selected.
--documents only | -d	Reindex all documents on the site that are currently not searchable or all artifacts if option f is selected.
---posts only	Reindex all posts on the site that are currently not searchable or all artifacts if option f is selected.
---trackers only	Reindex all trackers on the site.
---document_folders-only	Reindex all document folders on the site.
---topics-only	Reindex all topics.
---forums-only	Reindex all forums on the site.
---news-only	Reindex all news.
---project_pages-only	Reindex all project pages.
---packages	Reindex all packages.
---commits-only	Reindex all commits.
---frs_files-only	Reindex all frs files.
---releases-only	Reindex all releases.
---wikis-only	Reindex all wikis.
--project-id projectID | -p	Limit the re-indexing to data for single project when re-indexing only artifacts and or documents.
--verify | -x	Searches for each item that is scheduled for re-indexing. There is a one minute wait limit for each item to be re-indexed by the server.

--dryrun | -r	Executes all the steps for scheduling a re-index without actually sending any re-index requests to the server. This provides a list of items that need re-indexing.
output-file filePath, | -o	Prints the output for the given file.
--verbose | -v	Chronicles the process of scheduling the re-index a bit more.

set_auth_key.py

The `set_auth_key.py` script sets the authorized key to the scmviewer user profile.

Overview

The script accepts the authorized key file, reads the key from the file and sets it to the scmviewer user profile.

```
$ sudo /opt/collabnet/teamfoge/runtime/scripts/codesearch/set_auth_key.py --help
```

Usage

Run this script as follows:

```
./set_auth_key.py --authkey-file=<path_of_authkey_file>
```

When you run the script, you will be prompted to enter the TeamForge site-admin credentials to update the key for the scmviewer user.

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

wmt-wrapper.sh

The `wmt-wrapper.sh` script is used to invoke `wmt.java` by using `wmt.jar`. This script is a wrapper script for the `wmt.sh` script.

Overview

The wmt tool converts the CEE Moinmoin Wiki data to TeamForge JspWiki data. The `wmt-wrapper.sh` script is used to execute the wmt tool. For more information on WMTool, click *https://forge.collab.net/sf/wiki/do/viewPage/projects.sf_engine/wiki/WMTool*

Usage

Run this script for all projects:

```
$ <WMT_BUILD_DIR>/wmt-wrapper.sh -a
```

Run this script for a particular project:

```
$ <WMT_BUILD_DIR>/wmt-wrapper.sh -s.
```

For detailed information on running this script, click *Run the wmt-wrapper.sh script*

Options

Options that are available for this script:

-a | -- All projects	Runs the script for all the projects.
-s | -- Specific project	Runs the script for specific projects by mentioning the project name.

Log files in TeamForge 7.0 on Red HatCentOSSuSEVMware Player

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 7.0 on Red HatCentOSSuSEVMware Player

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/7.0.0.1/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 displayed in and exported from the **Planned Tracker Artifacts** tab available in **File Releases**.

Values

Integer

Default

5000

ARTIFACT_DESC_EDITOR

The *ARTIFACT_DESC_EDITOR* variable allows you to choose the type of text that can be used for artifact description using the editor tool.

Values

Rich Text/Plain Text

Default

Rich Text

Comments

You can use the RichText format only when you select the **WYSIWYG editor** from the Select Editor. This format is not supported when you choose the Plain Editor.

BDCS_ADMIN_USERNAME

The *BDCS_ADMIN_USERNAME* variable specifies the admin username to be created in the Black Duck Code Sight application.

Values

20 bytes, case-sensitive, alphanumeric, '_'. should begin with alpha

Default

sysadmin

BDCS_ADMIN_PASSWORD

The *BDCS_ADMIN_PASSWORD* variable specifies the password to be set in the Black Duck Code Sight application.

Values

15 Bytes, case-sensitive, alphanumeric, '_!@&*%'

Default

blackduck

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_FORUM_EDITOR

The *DISCUSSION_FORUM_EDITOR* variable allows you to choose the type of text that can be used in discussion forum description using the editor tool.

Values

Rich Text/Plain Text

Default

Rich Text

Comments

You can use the RichText format only when you select the **WYSIWYG editor** from the Select Editor. This format is not supported when you choose the Plain Editor.

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_POST_EDITOR

The *DISCUSSION_POST_EDITOR* variable allows you to choose the type of text that can be used for posting in discussion forums using the editor tool.

Values

Rich Text/Plain Text

Default

Rich Text

Comments

You can use the RichText format only when you select the **WYSIWYG editor** from the Select Editor. This format is not supported when you choose the Plain Editor.

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.

DOCUMENT_TEXT_EDITOR

The *DOCUMENT_TEXT_EDITOR* variable allows you to choose the type of text that can be used for the document description using the editor tool.

Values

Rich Text/Plain Text

Default

Rich Text

Comments

You can use the RichText format only when you select the **WYSIWYG editor** from the Select Editor. This format is not supported when you choose the Plain Editor.

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_BUILTIN_TOMCAT

The *ETL_BUILTIN_TOMCAT* variable specifies the type of Tomcat (internal or external) used for the ETL service.

Values

true or false

Default

true

Comment

If ETL_BUILTIN_TOMCAT=true, the internal Tomcat is used for the ETL service. If the token is set to false, it is mandatory to specify the home directory of the Tomcat's install directory in the token *EXTERNAL_TOMCAT_INSTALL_DIR*

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 30 2 * * ?

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

EXTERNAL_TOMCAT_INSTALL_DIR

The *EXTERNAL_TOMCAT_INSTALL_DIR* variable specifies the path to a valid Tomcat installation directory.

Values

Path specification.

Default

None

Comment

If either of these tokens, INTEGRATION_BUILTIN_TOMCAT/ETL_BUILTIN_TOMCAT are set to false, then it is mandatory to specify the home directory of the Tomcat install directory in the token EXTERNAL_TOMCAT_INSTALL_DIR

FORBIDDEN_PASSWORD

The *FORBIDDEN_PASSWORD* variable restricts specified words from being used as passwords.

Values

Comma-separated strings

Default

None

GERRIT_FORCE_HISTORY_PROTECTION

The *GERRIT_FORCE_HISTORY_PROTECTION* token determines whether the history protection feature of the TeamForge Git integration is enabled. If it is set to true, history protection is turned on for all repositories hosted on the Git integration server.

Values

true or false

Default

false

Comments

In TeamForge 7.0 (and later versions), this token is defined in the `runtime-option.conf` file to support the non-interactive installation of the Git integration. After Gerrit's post-install script is run, the value of this token is used to set the configuration property `forceHistoryProtection` in the `/opt/collabnet/gerrit/etc/gerrit.config` file.

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 link for forced password change.

Values

Integer (number of seconds)

Default

72

INTEGRATION_BUILTIN_TOMCAT

The *INTEGRATION_BUILTIN_TOMCAT* variable specifies the type of Tomcat (internal or external) used for SCM integrations.

Values

true or false

Default

true

Comment

If INTEGRATION_BUILTIN_TOMCAT=true, the internal Tomcat is used for SCM integrations. If the token is set to false, it is mandatory to specify the home directory of the Tomcat install directory in the token EXTERNAL_TOMCAT_INSTALL_DIR

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

JBOSS_ALARM_TIMEOUT

The *JBOSS_ALARM_TIMEOUT* variable specifies the time duration within which the JBoss service is expected to respond to requests sent by jboss_watchdog.

Values

Integer

Default

20

JUMP_TO_ID_ACTIVE_IN_QUICK_SEARCH

When the *JUMP_TO_ID_ACTIVE_IN_QUICK_SEARCH* variable is set to true, the **Jump to ID** quick search feature is replaced by the enhanced **Search** feature implemented in TeamForge 7.0 release.

Values

true or false

Default

true

Note: If you want to restore the legacy **Jump to ID** quick search, you must set this *JUMP_TO_ID_ACTIVE_IN_QUICK_SEARCH* variable to false in the `site-options.conf` file.

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_FILE

The *LOGIN_CONFIG_XML_FILE* 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

220

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

MIRROR DATABASE HOST

The *MIRROR_DATABASE_HOST* variable is a TeamForge database token that specifies the host of the database. This variable allows to extract the reporting data from the mirror TeamForge database through the Extract, Transform and Load (ETL) process.

Values

Alphanumeric string

Default

The MIRROR_ token takes the value of DATABASE_ token.

Comment

Example: Enter *MIRROR_DATABASE_HOST*=cu349.cloud.sp.collab.net (server name)

Add this token to the `site-options.conf` only if you setup a mirror database.

MIRROR_DATABASE_NAME

The *MIRROR_DATABASE_NAME* variable is a TeamForge database token that specifies the name of the TeamForge database. This variable allows to extract the reporting data from the mirror TeamForge database through the Extract, Transform and Load (ETL) process.

Values

Alphanumeric string

Default

The MIRROR_ token takes the value of DATABASE_ token.

Comment

Example: Enter *MIRROR_DATABASE_NAME*=ctfdb

Add this token to the `site-options.conf` only if you setup a mirror database.

MIRROR_DATABASE_PASSWORD

The *MIRROR_DATABASE_PASSWORD* variable is a TeamForge database token that specifies the password of the database. This variable allows to extract the reporting data from the mirror TeamForge database through the Extract, Transform and Load (ETL) process.

Values

Alphanumeric string

Default

The MIRROR_ token takes the value of DATABASE_ token.

Comment

Example: Enter *MIRROR_DATABASE_PASSWORD*=ctfpwd

Add this token to the `site-options.conf` only if you setup a mirror database.

MIRROR_DATABASE_PORT

The *MIRROR_DATABASE_PORT* variable is a TeamForge database token that specifies the port number of the database. This variable allows to extract the reporting data from the mirror TeamForge database through the Extract, Transform and Load (ETL) process.

Values

Port specification

Default

The MIRROR_ token takes the value of the DATABASE_ token.

Comment

Example: Enter *MIRROR_DATABASE_PORT*=5432.

Add this token to the `site-options.conf` only if you setup a mirror database.

MIRROR_DATABASE_USERNAME

The *MIRROR_DATABASE_USERNAME* variable is a TeamForge database token that specifies the database user's name. This variable allows to extract the reporting data from the mirror TeamForge database through the Extract, Transform and Load (ETL) process.

Values

Alphanumeric string

Default

The MIRROR_ token takes the value of the DATABASE_ token.

Comment

Example: Enter *MIRROR_DATABASE_USERNAME*=ctfuser

Add this token to the `site-options.conf` only if you setup a mirror database.

MODPAGESPEED_ENABLED

The *MODPAGESPEED_ENABLED* variable is used to enhance the performance of the web pages.

Values

true or false

Default

true

ORGANIZATION_EDITABLE

The *ORGANIZATION_EDITABLE* variable allows or prevents editing the organization value of a user account.

Values

true or false

Default

true

OBFUSCATION_ENABLED

The *OBFUSCATION_ENABLED* variable is used to run the TeamForge application in the obfuscation mode (if required) for security purpose.

Values

true or false

Default

false

Comment

When the TeamForge application is running in the obfuscation mode, the database login credentials, shared secrets etc., are encrypted and stored in the TeamForge configuration files for security reasons.

OBFUSCATION_KEY

The *OBFUSCATION_KEY* variable is used by the TeamForge obfuscation component as an input to the obfuscation algorithm for encryption and decryption purposes.

Values

AlphaNumeric (length greater than or equal to 8 bytes)

Default

XSJt43wN

OBFUSCATION_PREFIX

The *OBFUSCATION_PREFIX* variable is used as an identifier to determine whether the token variable is in obfuscated form.

Values

String (length between 4 to 8 bytes, can contain only the combination of following characters AlphaNumeric, _, {, }, %, $, #, @, -, [,])

Default

{OBF}:

Comments

The TeamForge obfuscation component prefixes the obfuscated value with the value of this variable before storing the obfuscated value in the configuration file.

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_CONTROL_EFFECTIVE_DATE

The *PASSWORD_CONTROL_EFFECTIVE_DATE* token is used to set the date from which the password security feature takes effect.

Values

Date (mm/dd/yyyy)

Default

Null

Comments

The *REQUIRE_PASSWORD_SECURITY* on page 381 is the master token that enables the password security feature.

Example 1

Consider a site with 130 users on which the password control kit (PCK) was not active. Of the 130 users, assume that:

- 100 users did not change password in last 100 days.
- 20 users did not change password in last 85 days.
- 10 users did not change password in last 75 days.

Assume that the following tokens are set on 01/01/2014 (current date):

- REQUIRE_PASSWORD_SECURITY=true
- PASSWORD_WARNING_PERIOD=20
- PASSWORD_EXPIRY_PERIOD=90
- PASSWORD_DISABLE_PERIOD=30
- PASSWORD_DELETE_PERIOD=60

PCK runs on 01/01/2014 and:

- If you have no PASSWORD_CONTROL_EFFECTIVE_DATE set:
  - 100 user accounts with no password change for the past 100 days would expire right away.
  - 20 users with no password change for the past 85 days would get a warning message that their passwords will expire in 5 days.
  - 10 users with no password change for the past 75 days would get a warning message that their passwords will expire in 15 days.
- If you have PASSWORD_CONTROL_EFFECTIVE_DATE=01/10/2014 (set to a future date):
  - 100 users with no password change for the past 100 days would get a warning message that their passwords will expire in 10 days.
  - 20 users with no password change for the past 85 days would get a warning message that their passwords will expire in 10 days.
  - 10 users with no password change for the past 75 days would get a warning message that their passwords will expire in 15 days.

Example 2

Consider the following scenario in which:

- Current date = 01/01/2014
- PASSWORD_CONTROL_EFFECTIVE_DATE=01/01/2013

In this scenario, the password control effective date is set to a date in the past. As a result, password control takes immediate effect and the PCK starts disabling, deleting or expiring user accounts right away.

PASSWORD_DELETE_PERIOD

The *PASSWORD_DELETE_PERIOD* variable specifies the time frame within which a disabled user account is automatically deleted.

Values

Integer (number of days)

Default

60

👉 **Note:** The *PASSWORD_DELETE_PERIOD* can be disabled by setting the value to zero.

PASSWORD_DISABLE_PERIOD

The *PASSWORD_DISABLE_PERIOD* variable specifies the time frame within which a user (soft-expired) is turned into a disabled user.

Values

Integer (number of days)

Default

30

Comments

A value of zero will disable this feature.

PASSWORD_EXPIRY_PERIOD

The *PASSWORD_EXPIRY_PERIOD* variable specifies the number of days after which the users' password expires.

Values

Integer (number of days)

Default

90

👉 **Note:** You cannot disable the *PASSWORD_EXPIRY_PERIOD* by setting the value to zero.

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

`135`

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_STATEMENT_TIMEOUT

The *PGSQL_STATEMENT_TIMEOUT* variable is set to prevent the Postgres queries from running for a long period of time.

Values

Integer (Milliseconds)

Default

600000 (Milliseconds)

Comments

An error message is displayed for every timeout in the postgres.log file and the log message with the exid id is logged in the vamessages.log and server.log files.

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

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

PLANNING_FOLDER_DISCUSSION_EDITOR

The *PLANNING_FOLDER_DISCUSSION_EDITOR* variable allows you to choose the type of text that can be used in the planning folder description using the editor tool.

Values

Rich Text/Plain Text

Default

Rich Text

Comments

You can use the RichText format only when you select the **WYSIWYG editor** from the Select Editor. This format is not supported when you choose the Plain Editor.

PLANNING_BOARD_SWIM_LANE_LIMIT

By default, not more than 250 cards are shown in a planning board swimlane. However, as a site administrator, you can increase or decrease the number of cards shown in the planning board swimlanes by configuring the site options token, PLANNING_BOARD_SWIM_LANE_LIMIT.

Values

A positive number.

Default

250

Comments

When you select a planning folder in one of the swimlanes and if X is greater than N, (where X = number of artifacts in the selected planning folder and N = PLANNING_BOARD_SWIM_LANE_LIMIT), the message, `Swimlanes in the Board View is currently configured to show N artifacts only`, appears at the bottom of the swimlane.

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* token, if set to true, enforces password security policy for the site.

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. You can also set the *PASSWORD_CONTROL_EFFECTIVE_DATE* token with a date from which

the password policy would be enforced. For more information, see *PASSWORD_CONTROL_EFFECTIVE_DATE* on page 372.

REQUIRE_RANDOM_ADMIN_PASSWORD

The *REQUIRE_RANDOM_ADMIN_PASSWORD* variable restricts users from setting a random admin password.

Values

true or false.

Default

True (SaaS), false (BTF)

Comment

This variable when set to true, checks for a valid mail id in the ADMIN_EMAIL token.

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.

SCM_SOAP_TIMEOUT

The *SCM_SOAP_TIMEOUT* variable is used to specify the connection timeout of the SCM soap requests between the APP and SCM servers.

Values

Integer (Milliseconds)

Default

300000 (Milliseconds)

SCM_USER_ENCRYPTED_PASSWORD

The *SCM_USER_ENCRYPTED_PASSWORD* variable is used to store the encrypted scmviewer password.

Values

- Alpha-numeric
- Special characters like '~!@#$%^&*'

Default

The default value will be in the encrypted format. See *password_util.sh* on page 339 for more information.

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.

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_CHAIN_FILE
The *SSL_CHAIN_FILE* variable specifies the path to the file where the TeamForge site's SSL certficate chain file 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_BROWSER_CACHE_PASSWORD

The *USE_BROWSER_CACHE_PASSWORD* variable restricts the storage of password in the browser when you login to the site.

Values

true/false

Default

true

USE_EXTERNAL_USER_AUTHENTICATION

The *USE_EXTERNAL_USER_AUTHENTICATION* variable specifies whether users can be authenticated through a separate system, such as OpenLDAP.

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_MONITORING_REMOVE_ENABLED

Set the *USER_MONITORING_REMOVE_ENABLED* variable to true, if you want to enable the feature that lets you remove one or more users from monitoring selected TeamForge objects.

Values

true or false

Default

false

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, scmviewer (for SaaS and BTF)

Comment

The variable is enabled by default and available in all the `site-options.conf` files.

Using multi-line blocks for site options

The multi-line block configuration is generally used by old SFEE sites. To define a `site-options.conf` token with a multi-line block value, you need to follow a certain syntax.

- Declare the token name with the value "START_MULTILINE_BLOCK". Syntax: <TOKEN_NAME>=START_MULTILINE_BLOCK

- Specify the multi-line values beneath the token.
- Complete the multi-line block with "END_MULTILINE_BLOCK" after all the multi-line values are specified. Syntax: END_MULTILINE_BLOCK

Example

```
SOURCEFORGE_CONFIGURATION_PROPERTIES_APPEND=START_MULTILINE_BLOCK
email.suppress.project_member_added=true
email.suppress.scm_user_password_synchronized=true
END_MULTILINE_BLOCK
```

c6migrate.conf variables

These are the configuration settings you can specify in the `c6migrate.conf` file.

Name	Description	Default	Values	Comments
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	The date-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.

A sample pebble-dep.xml file for the REST service type

```
<?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>https://cu064.cloud.sp.collab.net:13001/pebble/index.jsp</
baseurl>
    <gourl>https://cu064.cloud.sp.collab.net:13001/pebble/gourl/%p/%o</gourl>
    <endpoint>https://cu064.cloud.sp.collab.net:13001/pebble/services/rest/
ctfapi</endpoint>
 <servicetype>REST</servicetype>
</integrated-application>
```

A sample pebble-dep.xml file for the SOAP service type

```
<?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>https://cu177.cloud.maa.collab.net:13001/pebble/index.jsp</
baseurl>
    <gourl>https://cu177.cloud.maa.collab.net:13001/pebble/gourl/%p/%o</gourl>
<endpoint>https://cu177.cloud.maa.collab.net:13001/pebble/services/
PebbleIntegrationService</endpoint>
    <servicetype>SOAP</servicetype>
</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 version 6.1.1 and later, 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 the 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.

This tag is used in both deployment and application configuration files.

<adminurl>

When an application has an administration screen for configuring its parameters, this field contains that URL. It is optional.

This tag is used in the deployment configuration file.

<baseurl>

This is the URL to which a user will be directed on clicking the integrated application button in a project.

This tag is used in the deployment configuration file.

<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.

This tag is used in the deployment configuration file.

<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`.

This tag is used in the deployment configuration file.

<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.

This tag is used in the application configuration file.

<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.

This tag is used in the application configuration file.

<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.

This tag is used in the application configuration file.

<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)

This tag is used in the application configuration file.

<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.

This tag is used in the application configuration file.

<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. Click here for *steps to add integrated applications to a project*.

This tag is used in the application configuration file. The prefix can contain alpha-numeric characters and cannot be more than six characters in length. For more information about prefix, see *How does an integrated application interact with other TeamForge tools?*

<require-per-project-prefix>

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.

This tag is used in the application configuration file.

<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.

This tag is used in the application configuration file.

<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.

This tag is used in the application configuration file.

<servicetype>

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 and later support REST calls. The servicetype tag indicates whether the protocol used for communication is REST or SOAP.

This tag is used in the deployment configuration file.

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
```

install.conf

The `install.conf` file contains the data needed to manage the Review Board installer.

This is an example of a default (unedited) `install.conf` file. To create your own Review Board 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 Review Board installation directory
rb_dir=/u1/reviewboard
```

```
#Location of the Review Board data directory
rb_data_dir=/opt/collabnet/reviewboard/data

#Review Board site information
domain= myapp.collab.net
rb_database_type=postgresql
rb_database_name=ctfrbdb
rb_database_user=ctfrbuser
rb_database_password=ctfrbpwd
rb_database_host= <database_hostname>
rb_database_port=5432

#CTF Information
ctf_base_url=https://myapp.collab.net
ctf_site_var_dir=/opt/collabnet/teamforge/var
```

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
```

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>
```

CollabNet TeamForge 7.0 release notes

Last Revision: July 2013

Look here for information about changes in CollabNet software, such as new features, fixes and late-breaking updates.

New features in CollabNet TeamForge 7.0

TeamForge 7.0 adds these new features.

Note: For these new features, the tracking IDs, with the prefix "artf", are included within parenthesis.

Planning Board

The CollabNet TeamForge 7.0 introduces the new planning board. The planning board, with three swimlanes, is an important tool for your TeamForge project's Agile planning activities. When you've set up your planning folders, you have two views available to work with them, the List View and the new Board View (planning board). The planning board view enables you to plan and monitor the features that are required in each sprint (or iteration), and assign them from the product backlog to specific sprints. For more information, *click here*.

(artf139567)

By default, not more than 250 cards (artifacts) are shown in the planning board swimlanes. However, your site administrator can increase or decrease the number of cards shown in the planning board swimlanes by configuring the site options token, PLANNING_BOARD_SWIM_LANE_LIMIT.

(artf148005/artf146594)

Inline editing of tracker and planning folder artifacts

Inline editing of tracker and planning folder artifacts lets you update a list of artifacts at one stretch. As opposed to editing artifacts one-by-one, inline editing of multiple artifacts at once can save you time and effort. For more information, *click here*.

(artf119612)

Life cycle metric charts

You can add three new life cycle metric charts to your project pages. They are: Artifact Distribution Chart (Single Tracker), Artifact Distribution Chart (Multiple Trackers) and Artifact - Open/Close Chart (Multiple Trackers).

(artf128120)

TeamForge Orchestrate

The TeamForge Orchestrate acts as a control center for delivery pipelines, with social activity streams and open pluggable APIs. In other words, the TeamForge Orchestrate procures metadata about product lifecycle activities (such as SCM commits, builds, and code reviews), enables traceability between these activities and, further, associates these activities with TeamForge work items such as artifacts. For more information, refer to the TeamForge Orchestrate online help.

Usability improvement highlights

As a usability improvement, the count for the number of rows per page that can be viewed in tables is displayed in the label of the **Rows** drop-down list. This lets you know the number of rows displayed per page in a table without having to click the **Rows** drop-down list.

(artf138115)

The process of *applying, clearing, saving, and deleting filters* in TeamForge 7.0 has been further simplified.

(artf141261)

Always visible and sticky table header and footer: When you view a list of artifacts in the tracker List Artifacts view (for example, **Trackers** > **Stories** > **List Artifacts**) or the planning folder List Artifacts view and if you cannot see the table header as you scroll down, data has less meaning. Similarly, when you are at the top of the list and want to perform an action, you may have to scroll down to click the buttons located at the table footer. As a usability improvement, the table header and footer in the TeamForge 7.0 tracker List Artifacts view and planning folder List Artifacts view have been set to stay in sight even when you scroll up or down.

This feature has also been implemented in the **User-Role Matrix** tab used to assign roles to *project members* and *project group members*.

(artf132295/artf138116)

The **Jump to ID** quick search has been enhanced to let you search for TeamForge objects in a specific object category.

However, if for some reasons, you want to restore the legacy **Jump to ID** quick search, you can set the site options token *JUMP_TO_ID_ACTIVE_IN_QUICK_SEARCH* to false.

For more information on the enhanced **Search** feature, *click here*.

(artf141260/artf151879)

As a site or project administrator, you can remove one or more users from monitoring selected TeamForge objects so that no e-mail notifications are sent to the users about TeamForge objects that may not be of interest to them. However, you cannot remove a user from the monitoring list if the user is *monitoring applications* such as trackers, documents, tasks, and so on instead of individual TeamForge objects.

By default, this feature is disabled. To enable this feature, set the *USER_MONITORING_REMOVE_ENABLED* site options token to true.

(artf142274)

Password control kit (PCK) enhancement

The PCK is enhanced with a new token *PASSWORD_CONTROL_EFFECTIVE_DATE* to ensure robust password control enforcement. For more information, see *PASSWORD_CONTROL_EFFECTIVE_DATE* on page 372.

(artf144686)

TeamForge patch notification

Whenever a new TeamForge patch is available, site administrators are notified through a pop-up message in the site navigation bar.

(artf147528)

Reporting: SCM initial load job

Historical data collection job (initial load) for SCM commits is added in TeamForge 7.0.

The SCM initial load job is for collecting the historical SCM commit data. The SCM initial load job harvests all the commits made either as of the site installation date or the last successful incremental run, whichever is latest. Any commits after the last incremental run are harvested during the next incremental run. The SCM incremental job is blocked until the SCM initial load job is completed.

(artf143563/artf132678)

Review Board and GIT supported in SUSE installations

The TeamForge 7.0 SUSE (both 32 and 64 bit platform) installation supports integration of Review Board and GIT.

Event handler enhancement

A new event handling mechanism is implemented in the TeamForge 7.0 to improve the performance of the application. As a result, some background tasks are completed faster in large TeamForge instances or instances that process huge volume of information. In addition, sending emails and indexing of documents should be faster.

(artf97082)

View More and View Less links in tracker artifact description and comments

If a tracker artifact has a lengthy description or comment that exceeds 100,000 characters, you can find the **View More** link available in the description or comment. Clicking the **View More** link lets you view the description or comment incrementally in parts of 100,000 characters. You can also find the **View Less** link when you reach the end of a lengthy description or comment. Clicking **View Less** takes you to the beginning of the description or the comment with the **View More** link enabled again.

(artf130361)

Multiple prefixes for integrated applications

At times, you may want to use more than one prefix for an integrated application such as the TeamForge Orchestrate. It is possible to have multiple prefixes set up for integrated applications. You must have the prefixes, separated by commas, included in the XML application configuration file and upload the file to your TeamForge site. For more information, see *this FAQ topic*.

However, integrated applications such as the Project Tracker, Review Board, and Pebble do not support multiple prefixes.

(artf123825/artf149213)

Subversion upgrade

TeamForge 7.0 supports Subversion 1.7.11.

(artf153681)

Use of the Apache TIKA instead of the Stellent

The CollabNet TeamForge 7.0 uses the Apache TIKA parser library for indexing instead of the Stellent parser library, to detect and extract metadata and structured text content from documents. For more information, refer to *Advantages of using the Apache TIKA parser library for indexing* on page 313.

(artf160669)

Issues resolved in CollabNet TeamForge 7.0

TeamForge 7.0 resolves these issues.

Note: For these resolved issues, the tracking IDs, with the prefix "artf", are included within parenthesis.

IAF

As a user with sufficient permissions, you can create a relationship between the TeamForge and an integrated application object. While other users without sufficient permissions on the integrated application were allowed to view the relationship, they cannot access the object and they used to get a `Permission Denied` error message. In TeamForge 7.0 release, this behavior has been changed so that users cannot view the objects for which they do not have sufficient permissions.

(artf141265)

Ad hoc database query restriction for `sfuser` and `password_history` tables

For security reasons, the `sfuser` and `password_history` tables of the operational datastore are restricted for ad hoc querying from the **Admin** > **System Tools** > **Ad Hoc Database Query** page. Alternatively, use the following views, `sfuser_view` and `password_history_view`, for retrieving all other data but passwords.

(artf125835)

Case sensitivity of integrated application object IDs

Integrated application object IDs are of the form "`prefix_objectId`". In an object ID such as "`prefix_objectId`", the "`prefix`" is case-insensitive, whereas the "`objectId`" is case-sensitive. For example, the two object IDs, "`PT_SC1`" and "`pt_SC1`", refer to the same object in TeamForge. Whereas, the two object IDs, "`PT_SC1`" and "`PT_sc1`", refer to two different objects in TeamForge. Here, `PT` and `pt` are case-insensitive and the `SC1` and `sc1` are case-sensitive. For more information, see *How does an integrated application interact with other TeamForge tools?*

(artf140407)

Known issues in CollabNet TeamForge 7.0

The following issues, including any workarounds we may have, are known to exist in the TeamForge 7.0 release.

Note: For these known issues, the tracking IDs, with the prefix "artf", are included within parenthesis.

Install / Upgrade

The TeamForge - Git Integration is not supported in advanced mode installation.

(artf130223)

The TeamForge installer starts the Apache server (restarts if the Apache server is running already) during create runtime in all the installation modes such as SaaS, dedicated and advanced.

(artf148234)

If you upgrade from TeamForge 6.2 or earlier to TeamForge 7.0 with unprocessed events in the `stored_message` table, few of those events will be ignored post upgrade. This is due to the event incompatibility issue. This issue is not applicable if you upgrade from TeamForge 6.2 Patch 1 to TeamForge 7.0.

(artf148628)

In SuSE advanced mode installation, the create-runtime fails while running the 99-z-teamforge-add-ons.py script as it fails to restart the Apache service. This can be ignored.

(artf152267)

Enable datamart to create life cycle metric charts

You must have datamart enabled on your site to create life cycle metric charts. If datamart is not enabled, an error message is displayed when you select the **Lifecycle Metrics** option on the **Create Component** page.

(artf152635)

Browser

When you upgrade from IE 7 or IE 8 to IE 9, you must ensure the Internet Explorer's **Document Mode** is set to IE 8 standards or IE 9 standards, failing which some of the TeamForge application's user interface controls may look distorted. To set your Internet Explorer's **Document Mode**:

1. Select **Tools** > **F12 developer tools**. The **developer tools** pane appears.
2. Select **Internet Explorer 8 standards** or **Internet Explorer 9 standards (Page default)** from the **Document Mode** menu.

(artf139944)

Additional RAM recommendation for TeamForge 7.0

As a result of significant infrastructure and platform changes for TeamForge 7.0 (JBoss 7.0, Postgres 9.0.13, Tika, Red Hat Linux 6.3 and so on), it is highly recommended that you add more RAM, which is to be exclusively allocated to Postgres. CollabNet recommends adding 50% more RAM for 7.0 compared to TeamForge 6.2. Contact the CollabNet Support for tuning the application for the additional RAM.

(artf150011)

Ad hoc database query restriction for `sfuser` and `passowrd_history` tables

For security reasons, the `sfuser` and `password_history` tables of the operational datastore are restricted for ad hoc querying from the **Admin** > **System Tools** > **Ad Hoc Database Query** page. Alternatively, use the following views, `sfuser_view` and `password_history_view`, for retrieving all other data but passwords.

(artf125835)

Usage of the term "REV" in commit messages

The term "REV", followed by a space and a number, if used in your commit message, becomes a hyperlink. For example, "REV 1". This is because, the term "REV" is a reserved keyword. However, if you click the hyperlink, the following error message appears: "`Could not authenticate user for SCM viewer`". It is recommended not to use the keyword "REV" followed by a space and a number in your commit messages.

(artf149518)

History > My Recently Edited menu

Using Planning Board, if you move an artifact from one planning folder (source swimlane) to the other (destination swimlane), you cannot find the recently moved artifact on the **History** > **My Recently Edited** menu if the artifact has been either:

- Placed as the bottommost artifact in the destination swimlane

  Or
- Moved to an "empty" planning folder (destination swimlane).

(artf146995)

Inline editing of the estimated, remaining and actual effort of an artifact

What's the objective?

Use inline editing to update the estimated, remaining, and actual effort of an artifact for which the automatic effort calculation is enabled.

What's the assumption?

Assume that you have enabled the automatic effort calculation for an artifact, say "artf1001", by selecting the **Calculate Effort: Sum effort from children** check box.

What are the steps to edit the effort inline?

1. Go to the tracker's **List Artifacts** page.
2. Configure the columns to include the estimated, remaining and actual effort columns.
3. Click **Edit Inline**.
4. Edit the estimated, remaining, and actual effort of the artifact, which in this case is "artf1001".
5. Click **Save**.

What's the result?

Though the application lets you edit the estimated, remaining and actual effort of "artf1001", the changes are not saved as the automatic effort calculation is enabled for "artf1001". This is a known issue.

(artf151224)

Inline editing errors

Assume that three or more tracker fields of type "Single Select" are created and hierarchically configured such that "field1" becomes the parent of "field2", "field2" becomes the parent of "field3" and so on. If you try to inline-edit any of the above fields (field1, field2, field3, etc), a Javascript error pops up. However, you can ignore the Javascript error, edit the drop-down fields and save the changes.

Similarly, assume that two or more fields of type "Single Select" are created and hierarchically configured in the same fashion as discussed earlier and that you are using the Firefox browser. If you try to inline-edit any of the hierarchically configured fields, the bottommost value in the drop-down is shown by default instead of "None".

(artf152635)

With some of the configurable fields such as Group, Category, Customer, Fixed in Release and so on, if "None" is the only available value in the drop-down list, an error message appears when you try to edit the fields inline and save them.

(artf152635)

When you edit artifacts inline, you may witness some parsing issues with fields such as "Fixed in Release" and "Reported in Release".

(artf152635)

Page drifts during inline editing

During inline editing of artifacts, you may witness random up or down page movements the first time when you click a field (that has a hyperlink) in some of the inline-editable columns such as the "**Assigned To**" column. Though this behavior is known to exist, it may not consistently recur at all times.

(artf151286)

Tasks with pre-1970 start and end dates

When you try to edit tasks with pre-1970 start and end dates, the **Edit Task** page shows no start and end dates. This is a known issue.

(artf151661)

CollabNet TeamForge 7.0 Patch 1 release notes

Release date: 11 Nov, 2013.

Look here for information about TeamForge 7.0 Patch 1, such as new features, fixed issues and late-breaking updates.

Apply TeamForge 7.0 Patch 1

Note:

TeamForge add-on packages can remain installed when a patch for TeamForge 7.0 is applied. No updates are required. However, you should verify the compatibility of Add-ons applied to your site as some may be more specific to your site.

The most common Add-ons and their TeamForge compatibility are published here: *https://ctf.open.collab.net/sf/go/page1363*

If you have any questions regarding compatibility or updating a specific add-on, contact *CollabNet Support* for more details.

If you have TeamForge 7.0 and are applying this patch, see the specific instructions for your operating system:

- *Apply a patch on RHEL / CentOS* on page 226
- *Apply a patch on SuSE* on page 228
- *Apply a patch on VMware Player or VMware ESXi* on page 230

New features
Subversion 1.7.11 TeamForge 7.0 Patch 1 supports Subversion 1.7.11. (artf158308)

Fixed issues
E-mails from unmonitored discussion forums Some users received random emails from discussion forums they did not monitor. This issue has been fixed. (artf158028) **LDAP: Issue with the createUser method's soap60 call** An issue with the createUser method's soap60 call was found, which allowed successful user creation regardless of whether the external authentication succeeds or not. This issue has been fixed. As part of the fix, if you have LDAP set up for external authentication, you must set the "REQUIRE_USER_PASSWORD_CHANGE" site options token to "false". For more information, refer to: • *Set up LDAP integration for the CollabNet TeamForge site* on page 235 • *REQUIRE_USER_PASSWORD_CHANGE* on page 382 (artf158029) **Discussion forum: Attachments in Simbel-imported discussion posts inaccessible** When a discussion forum was imported via Simbel, the URL for the posts in the discussion forum was having the "topic name" instead of the "topic ID". As a result, if a Simbel-imported post has an attachment, it became inaccessible. This issue has been fixed.

Fixed issues
(artf158031) **E-mail delivery issues with thread locks** With the introduction of new timeout tokens, the application server now times out when trying to send emails via a busy or down E-mail server. This used to hang forever and affected the E-mail delivery queue. 👉 **Important:** You must run the **post-migrate.py** script post upgrade. For more information, refer to *Apply a patch on RHEL / CentOS* on page 226. (artf158038) **SCM: Duplicate commit messages** The soap response of the ScmAppSoap.getCommitData() call was duplicating the commit message for every file included in a commit operation. As a result, there was an out of memory error (OOME) when a commit object with large number of files is accessed via soap using the getCommitData() method. Usage of getCommitData2() method resolves this issue and it is recommended to use getCommitData2() instead of getCommitData(). (artf158035) **Logging Issues in TeamForge 7.0** The profiler information such as page start or end id and session context information such as username and jsession id were found to be missing in the log files. This issue has been fixed. (artf160195) **TeamForge project creation issue** Some rolled back transactions resulted in duplicate sequence ids, which led to project creation failures. This issue has been fixed. (artf158039)