

# TeamForge 7.1 Administration Guide



# Contents

<b>TeamForge system administrator how-tos.....</b>	<b>6</b>
Install CollabNet TeamForge 7.1.....	6
Plan your CollabNet TeamForge 7.1 installation.....	6
Set up networking for your TeamForge server.....	7
Install CollabNet TeamForge 7.1 on Red Hat/CentOS.....	7
Install CollabNet TeamForge 7.1 on SuSE.....	58
Install TeamForge on VMware Player or VMware ESXi.....	107
Upgrade to TeamForge 7.1.....	111
Plan your upgrade to TeamForge 7.1.....	111
Upgrade to TeamForge 7.1 on Red Hat/CentOS.....	112
Upgrade to TeamForge 7.1 on SuSE.....	182
Upgrade to CollabNet TeamForge 7.1 on a virtual machine.....	251
Is my TeamForge site "dedicated" or "advanced?".....	253
Enable reporting while upgrading from TeamForge 6.1.1 to 7.1.....	254
Troubleshooting: Upgrade PostgreSQL manually.....	254
Install a different build of TeamForge 7.1.....	255
Maintain your TeamForge 7.1 site.....	256
Upgrade PostgreSQL using PGTurant.....	256
Upgrade CLI reports to the latest version on Red Hat, CentOS or SUSE.....	258
Install Indexer on a separate server.....	258
Supply your TeamForge license key from Teamforge user interface.....	260
Supply your CollabNet TeamForge license key as a text file.....	260
Support CollabNet TeamForge System Administrators.....	261
Protect your CollabNet TeamForge site.....	270
Get information about a CollabNet TeamForge site.....	276
Rebuild runtime without the install directory.....	276
Turn on site-wide reporting.....	277
Synchronize TeamForge source control integrations.....	277
Provide more than one source control server.....	278
Upgrade Subversion on RedHat or CentOS.....	278
Upgrade Subversion on SuSE.....	279
Change the scmviewer password.....	279
Change your site's domain name.....	280
Specify DNS servers.....	281
Optimize PostgreSQL with vacuum.....	281
Change the location of a log file.....	281
Change the logging level on your site.....	281
Raise the logging visibility of selected database requests.....	282
Rotate TeamForge log files.....	283
Schedule data extraction for reporting.....	283
Removing users from monitoring objects.....	284
Back up and restore CollabNet TeamForge data.....	284
Move the datamart to a separate box.....	286
Integrate TeamForge 7.1 with other tools.....	288
Integrated application example: Pebble.....	288
Set up Black Duck Code Sight.....	289
Set up Review Board.....	305
Set up Git.....	316
A preface to the TeamForge-Orchestrate integration.....	317

## Frequently asked questions about administrating a TeamForge site..... 318

TeamForge installation/upgrade FAQs.....	318
TeamForge 7.1 installation/upgrade FAQs.....	318
Users invoked via su command in TeamForge.....	319
Do I need an advanced TeamForge installation?.....	319
How many servers do I need to run a CollabNet TeamForge site?.....	320
What are the right PostgreSQL settings for my site?.....	321
How does TeamForge handle third-party applications?.....	321
Should I upgrade to TeamForge 7.1 on a new box?.....	322
Should I move my TeamForge database to its own server?.....	323
Should I move my source control application to its own server?.....	323
Is it possible to change artifact prefixes in TeamForge?.....	323
Can I run other java applications in the same JBoss instance as CollabNet Team Forge?.....	323
What does it mean to run CollabNet TeamForge on a virtual machine?.....	324
Who is responsible for applying OS updates to the underlying VMware image?.....	324
What is a patch?.....	324
Why do I get a URL "not found" or "moved permanently" error after applying a patch/upgrade?.....	324
Why does the the Yum installer display a warning message on Centos 6?.....	325
Why am I getting a Yum repository filename conflict?.....	325
When do I run the initial load job?.....	325
Common errors in TeamForge.....	326
Troubleshooting VMware errors.....	326
Troubleshooting database/datamart/ETL errors.....	327
Troubleshooting JBoss errors.....	329
Troubleshooting E-mail errors.....	329
Troubleshooting other errors.....	331
How does TeamForge manage security?.....	333
What are the minimum ports to keep open for a TeamForge site?.....	333
How does CollabNet TeamForge help protect data access?.....	335
What user activities are tracked?.....	336
How does CollabNet TeamForge help protect my data?.....	336
J2EE Architecture and security.....	336
What security tools come with CollabNet TeamForge ?.....	337
What is a CERT advisory?.....	338
How does TeamForge authenticate CVS users?.....	338
How do I configure Subversion to authenticate against multiple LDAP domains?.....	339
How do I authenticate multiple LDAP via Apache?.....	340
After switching to ADS authentication, why did the Create button disappear from the user admin section?.....	340
Does TeamForge work with LDAP?.....	341
Why do I get the "Invalid command 'AuthLDAPAuthoritative'" error when I try to set LDAP for SVN users?.....	342
How does TeamForge handle multiple redundant LDAP servers?.....	342
Can the users be forced to change their passwords at first login?.....	343
Managing email in TeamForge.....	343
How do I configure TeamForge to send mail on a specific network adapter in a multi-NIC configuration?.....	343
How can I check if port 25 is open?.....	343
How do I set up a local alias via James?.....	343
How do I configure email notifications of Subversion commits in SourceForge 4.x?.....	344
Does TeamForge support using /etc/aliases for local mail delivery?.....	344
Concepts and terms in TeamForge.....	344
Advantages of using the Apache TIKa parser library for indexing.....	344
What is the look project?.....	345
What wiki engine does TeamForge use?.....	345
Does CollabNet TeamForge support merge tracking?.....	345
What is a private IP address and what are the private IP ranges?.....	346

What is the vessages.log used for?.....	346
How do I use the TeamForge updater to manage backups of old versions of TeamForge?.....	346
How does TeamForge deliver activity reports?.....	346
What is an integrated application?.....	347
How does an integrated application interact with other TeamForge tools?.....	347
TeamForge roles and permissions.....	348
Can I set permissions so that users can move documents but not delete them?.....	349
Why can't Oracle connect to my TeamForge installation?.....	349
Are role-based permissions allowed for sub-folders in the TeamForge Documents?.....	349
Can I control user access to an integrated application?.....	349
Tasks in TeamForge.....	349
How do I change the time to run the ETL jobs?.....	350
How can I check the status of ETL?.....	350
What happens when log files get too big?.....	350
What is the suggested log configuration for a production system?.....	350
How do I enable post-commit logging?.....	350
How do I make the monitoring messages be sent from Forge Administrator?.....	350
How can I remove the RHEL test page after TeamForge installation?.....	350
How to reinstall a deleted installation directory?.....	351
How can I find the number of files in a repository without checking it out?.....	351
How do I connect to the Datamart?.....	351
How do I connect to the Teamforge Postgres database?.....	351
How do I generate a wiki table of contents?.....	351
What is the correct procedure for modifying a hosted Lab Manager profile?.....	351
How do I configure the timeout for Apache in TeamForge?.....	352
How do I back up TeamForge?.....	352
How do I move an existing CVS repository into TeamForge?.....	353
How do I move an existing SVN repository into TeamForge?.....	353
Where do I configure my client proxy settings?.....	354
How do I make TeamForge work the same when the IP address of the server changes?.....	354
How do I capture the output of "top" command?.....	355

## **Reference information about TeamForge..... 356**

Platform specification for TeamForge 7.1.....	356
Hardware requirements for CollabNet TeamForge 7.1.....	356
Software requirements for CollabNet TeamForge 7.1.....	356
Versions of RPM packages in Red Hat and CentOS installations - TeamForge 7.1.....	358
Versions of RPM packages in SUSE installations - TeamForge 7.1.....	359
Hardware and software requirements for CollabNet TeamForge 7.1 on a virtual machine.....	360
Scripts installed with TeamForge.....	360
<b>backup-rb-data.py</b> .....	360
bootstrap-data.sh.....	360
bootstrap-reporting-data.sh.....	361
CodeSightMigration.sh.....	362
The collabnet script.....	364
datamart-oracle-setup.sh.....	365
datamart-pgsql-setup.sh.....	365
db.py.....	365
domain_change_db.py.....	366
domain_change_fs.pl.....	366
domain_change_pt.py.....	367
environment_check.sh.....	367
etl-Client.py.....	368
fix_data_permission.sh.....	368
install.sh.....	369

pbl.py.....	370
password_util.sh.....	371
psql-wrapper.....	371
psql-reporting-wrapper.....	372
<b>restore-data.py</b> .....	372
SearchReindex.py.....	373
set_auth_key.py.....	374
set-reports-readonly-user-permission.py.....	374
snapshot.py.....	375
upgrade-site.sh.....	375
projecttracker.py.....	377
wmt-wrapper.sh.....	378
TeamForge 7.1 scripts.....	378
Log files in TeamForge - Red HatCentOSSuSEVMware Player.....	379
JBoss logs.....	379
Oracle logging.....	379
SCM (CVS, Subversion, and Perforce) logs.....	380
Email logs.....	380
Search logs.....	381
Project Build Library audit log.....	381
Profile audit log.....	382
User Audit Log.....	382
Host audit log.....	382
Project audit log.....	383
etl.log.....	383
Configuration files in TeamForge - Red HatCentOSSuSEVMware Player.....	383
site-options.conf.....	383
c6migrate.conf variables.....	421
httpd.conf.....	422
pebble-app.xml.....	423
pebble-dep.xml.....	425
How is an integrated application described?.....	425
install.conf.....	429
install.conf.....	429
iptables.....	430
login-config.xml.....	430
The patch manifest file.....	431
postresql.conf.....	432

## **CollabNet TeamForge 7.1 release notes..... 434**

New features in CollabNet TeamForge 7.1.....	434
Issues resolved in CollabNet TeamForge 7.1.....	439
Known issues in CollabNet TeamForge 7.1.....	441

# TeamForge system administrator how-tos

---


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

## Install CollabNet TeamForge 7.1

---

When you finish the sequence of tasks given below, you will have a working TeamForge 7.1 site tailored to the specific requirements of your user base and environment.

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

 **Note:** For the hardware and software required to run TeamForge, see [Platform specification for TeamForge 7.1](#) on page 356.


## Plan your CollabNet TeamForge 7.1 installation

Before you install TeamForge 7.1, let's take a look at the product from a system administrator's perspective, so that you know exactly what you are getting into.

### Overview

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

- The core TeamForge application provides the Web interface that users see, and the API that other applications can interact with. It also includes the file system where some user content is stored, such as wiki pages.
- The site database is where most of the user-created content is stored and accessed. Documents, discussion posts, tracker artifacts, project administration settings: all that sort of thing lives in the database.
- The source control server ties any number of Subversion, CVS or Perforce repositories into the TeamForge site.
- 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](mailto: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 server as the main TeamForge application. But in practice, only the TeamForge application *needs* to run on the TeamForge application server. The other services can share that server or run on other servers, in almost any combination. When you distribute your services on multiple servers, 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 9.2.4 is installed automatically when you install TeamForge 7.1. 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 7.1](#) on page 356.

## Set up networking for your TeamForge server

After installing the operating system, prepare the networking connections and configuration that your TeamForge 7.1 site will require.

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

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

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


For detailed port requirements, see [What are the minimum ports to keep open for a TeamForge site?](#) on page 333

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 333 for detailed security recommendations.)

## Install CollabNet TeamForge 7.1 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 7.1 with all services on the same server

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

In this option, the following services run on the application server (we call this my.app.host).

- TeamForge Application Server
- Black Duck Code Sight Server
- Database Server (Operational DB and Reports DB)
- ETL Server
- GIT Integration Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer).

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

1. Install Red Hat Enterprise Linux / CentOS 6.4 or later versions and log in as root.

- The host must be registered with the Red Hat Network if you are using Red Hat Enterprise Linux. See [Platform specification for TeamForge 7.1](#) on page 356 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 server](#) on page 7 for details.

3. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.

c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```

See [Set up SELINUX](#) on page 271 to have TeamForge to run in SELinux mode after completing the installation or upgrade.

4. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

5. 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) 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/7.1.0.0/conf/site-options.conf
```



- a) Configure the HOST token.

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

```
DOMAIN_localhost=my.app.domain.com
```


- b) Configure the following settings if you are installing Git.

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

- c) Configure the following settings if you are installing Black Duck Code Sight.

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

- d) Configure the database and datamart settings.

 **Note:** For more information about configuring variables, see [site-options.conf](#) on page 383

```
DATABASE_TYPE=postgresql
```

```
DATABASE_USERNAME=ctfuser
```

```
DATABASE_NAME=ctfdb
```

```
DATABASE_READ_ONLY_USER=ctfrouser
```

```
REPORTS_DATABASE_USERNAME=ctfrptuser
```

```
REPORTS_DATABASE_NAME=ctfrptdb
```

```
REPORTS_DATABASE_READ_ONLY_USER=ctfrptrouser
```

```
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

 **Note:** The database name and username values are arbitrary alphanumeric strings.

- e) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to **\$auto\$**, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- DATABASE\_PASSWORD
- DATABASE\_READ\_ONLY\_PASSWORD
- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD
- BDCS\_ADMIN\_PASSWORD
- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


- f) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSSION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSSION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSSION_KEY` token is `XSJt43wN`.

To configure the `OBFUSSION_PREFIX` on page 405, set `OBFUSSION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSSION_PREFIX` is `{OBF}`).

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- g) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

```

• SSL=on
• SSL_CERT_FILE=
• SSL_KEY_FILE=
• SSL_CA_CERT_FILE=
• SSL_CHAIN_FILE=

```

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- h) If the token `REQUIRE_PASSWORD_SECURITY` is enabled, then set a value for the token, `PASSWORD_CONTROL_EFFECTIVE_DATE`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- i) 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.
- j) 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__}
```
- k) If you have LDAP set up for external authentication, you must set the `"REQUIRE_USER_PASSWORD_CHANGE"` site options token to false.
- l) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and PostgreSQL automatically.
- m) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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

If you are installing Git, 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
```

- n) Make sure that the following tokens have a value if ETL is enabled.

```
SOAP_ANONYMOUS_SHARED_SECRET=
```


```
ETL_SOAP_SHARED_SECRET=
```

- o) 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


-  **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 username add this token:  
BDCS\_ADMIN\_USERNAME=<sysadmin>  
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 are 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 the 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

- p) To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396

- q) If you are installing TeamForge through disconnected media, set the token `HELP_AVAILABILITY=local`.
- r) Ensure to set the token `SELINUX_SETUP=false` temporarily in the `site-options.conf` file.
- s) Save the `site-options.conf` file.

7. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

8. If you are installing on a server that is behind a proxy, unset the `http_proxy` token.

```
export http_proxy=
```

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

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

10. 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.

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

12. 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 302.

13. If you have installed Git, integrate gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

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
```

14. Run the following initial load jobs (ETL).

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 325.

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

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

16. 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.

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

17. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

18. Restart the collabnet services.

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

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

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

- c) Create a sample project.

See [Create a TeamForge project](#).

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

See [Create a site-wide broadcast](#).

For specific instructions on installing Review Board, see: [Set up Review Board](#) on page 305

To run TeamForge in SELINUX enabled mode, see [Set up SELINUX](#) on page 271

## Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see [TeamForge Orchestrate installation](#).

### Install TeamForge 7.1 with Database and Datamart on a separate server

In this option, we install the Database (Operational Database) and Datamart (Reporting database) on a separate server and other services on the main application server.


In this option, the following services run on the application server (we call this `my.app.host`).

- TeamForge Application Server
- Black Duck Code Sight Server
- ETL Server
- GIT Integration Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer).

The following service runs on the database server. (We call this `my.db.host`)

- Database Server (Operational DB and Reports DB)

We call this a "dedicated install" since both the servers are dedicated to TeamForge. The TeamForge 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.

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

Log in to the server as root.

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

1. Install Red Hat Enterprise Linux / CentOS 6.4 or later versions and log in as root.

- The host must be registered with the Red Hat Network if you are using Red Hat Enterprise Linux. See [Platform specification for TeamForge 7.1](#) on page 356 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 server](#) on page 7 for details.

3. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

- a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

- b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.

- c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```

See [Set up SELINUX](#) on page 271 to have TeamForge to run in SELinux mode after completing the installation or upgrade.

4. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

5. Install the following application packages.

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

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

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

```
yum install teamforge-git
```

- c) 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/7.1.0.0/conf/site-options.conf
```

- a) Identify the servers and services running on them.

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

```
DOMAIN_localhost=my.app.domain.com
```

```
HOST_<my.db.domain.com>=database datamart
```


- b) Configure the following settings if you are installing Git.

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

- c) Configure the following settings if you are installing Black Duck Code Sight.

```
HOST_localhost=app etl indexer subversion cvs gerrit codesearch
```

- d) Configure the database and datamart settings.

 **Note:** For more information about configuring variables, see [site-options.conf](#) on page 383

```
DATABASE_TYPE=postgresql
```

```
DATABASE_USERNAME=ctfuser
```

```
DATABASE_NAME=ctfdb
```


```
DATABASE_READ_ONLY_USER=ctfrouser
```

```
REPORTS_DATABASE_USERNAME=ctfrptuser
```

```
REPORTS_DATABASE_NAME=ctfrptdb
```

```
REPORTS_DATABASE_READ_ONLY_USER=ctfrptrouser
```

```
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

 **Note:** The database name and username values are arbitrary alphanumeric strings.

- e) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to **Sauto\$**, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- DATABASE\_PASSWORD
- DATABASE\_READ\_ONLY\_PASSWORD
- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD
- BDCS\_ADMIN\_PASSWORD
- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


- f) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the [OBFUSCATION\\_PREFIX](#) on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF};`.

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- g) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

```

• SSL=on
• SSL_CERT_FILE=
• SSL_KEY_FILE=
• SSL_CA_CERT_FILE=
• SSL_CHAIN_FILE=

```

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- h) If the token `REQUIRE_PASSWORD_SECURITY` is enabled, then set a value for the token, `PASSWORD_CONTROL_EFFECTIVE_DATE`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- i) 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.
- j) 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__}
```
- k) If you have LDAP set up for external authentication, you must set the `REQUIRE_USER_PASSWORD_CHANGE` site options token to false.
- l) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and Postgresql automatically.
- m) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

- n) Make sure that the following tokens have a value if ETL is enabled.

```
SOAP_ANONYMOUS_SHARED_SECRET=
ETL_SOAP_SHARED_SECRET=
```

- o) 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
```



- 👉 **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 username add this
token:
BDCS_ADMIN_USERNAME=<sysadmin>
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 are 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 the 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
```

- p) To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
  - q) If you are installing TeamForge through disconnected media, set the token **HELP\_AVAILABILITY=local**.
  - r) Ensure to set the token **SELINUX\_SETUP=false** temporarily in the `site-options.conf` file.
  - s) Save the `site-options.conf` file.
7. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

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

```
export http_proxy=
```

**Do this on the database server - my.db.host**

9. Install Red Hat Enterprise Linux / CentOS 6.4 or later versions and log in as root.

- The host must be registered with the Red Hat Network if you are using Red Hat Enterprise Linux. See [Platform specification for TeamForge 7.1](#) on page 356 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. Check your basic networking setup.

See [Set up networking for your TeamForge server](#) on page 7 for details.

11. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.

c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```


12. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

13. Install the TeamForge database packages.

```
yum install teamforge-database
```

14. Copy the `site-options.conf` file from the `my.app.host` to the database server in the directory `/opt/collabnet/teamforge-installer/7.1.0.0/conf`

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

 **Note:** If you choose not to use the application server's `site-options.conf` file, then don't forget to copy the value of `AUTO_DATA` token from the application server.

```
HOST_localhost=database datamart
```

```
DOMAIN_localhost=my.db.domain.com
```

```
HOST_my.app.domain.com=app etl indexer subversion cvs gerrit codesearch
```

16. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```


**Do the following on the application server - my.app.host**

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

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./bootstrap-data.sh
```


18. Start TeamForge.

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

 **Note:** Startup can take several minutes, depending on the speed of the host system configuration. 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.

19. If you have installed GIT, create the 'gitadmin' user (which is already added in the site-options token `USERS_WITH_NO_EXPIRY_PASSWORD`) with site administrator rights in the TeamForge user interface.

20. 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 302.

21. If you have installed Git, integrate gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

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
```

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

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

23. Run the following initial load jobs (ETL).

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 325.

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.
- 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. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

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

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

27. 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
```

28. 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
```

29. Restart the collabnet services.

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

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

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

b) 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.

c) Create a sample project.

See [Create a TeamForge project](#).

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

See [Create a site-wide broadcast](#).

For specific instructions on installing Review Board, see: [Set up Review Board](#) on page 305


To run TeamForge in SELINUX enabled mode, see [Set up SELINUX](#) on page 271

### Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see [TeamForge Orchestrate installation](#).

### Install TeamForge 7.1 with Reporting services on a separate server

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

 **Note:** For the ETL service to run as expected in a distributed TeamForge installation, all servers must have the same time zone.

In this option, the following services run on the application server (we call this `my.app.host`).

- TeamForge Application Server
- Black Duck Code Sight Server
- Database Server (Operational DB)
- GIT Integration Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer).

The following service runs on the database server. (We call this `my.reports.host`)

- Database Server (Reports DB)
- ETL Server

Log in to the server as root.

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

1. Install Red Hat Enterprise Linux / CentOS 6.4 or later versions and log in as root.

- The host must be registered with the Red Hat Network if you are using Red Hat Enterprise Linux. See [Platform specification for TeamForge 7.1](#) on page 356 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 server](#) on page 7 for details.

3. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

- a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

- b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.

- c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```

See [Set up SELINUX](#) on page 271 to have TeamForge to run in SELinux mode after completing the installation or upgrade.

4. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

5. Install the following application packages.

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

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

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

```
yum install teamforge-git
```

- c) 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/7.1.0.0/conf/site-options.conf
```

- a) Identify the servers and services running on them.

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

```
DOMAIN_localhost=my.app.domain.com
```

```
HOST_<my.reports.domain.com>=datamart etl
```


- b) Configure the following settings if you are installing Git.

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

- c) Configure the following settings if you are installing Black Duck Code Sight.

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

- d) Configure the database and datamart settings.

 **Note:** For more information about configuring variables, see [site-options.conf](#) on page 383

```
DATABASE_TYPE=postgresql
```

```
DATABASE_USERNAME=ctfuser
```

```
DATABASE_NAME=ctfdb
```


```
DATABASE_READ_ONLY_USER=ctfrouser
```

```
REPORTS_DATABASE_USERNAME=ctfrptuser
```

```
REPORTS_DATABASE_NAME=ctfrptdb
```

```
REPORTS_DATABASE_READ_ONLY_USER=ctfrptrouser
```

```
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

 **Note:** The database name and username values are arbitrary alphanumeric strings.

- e) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to **Sauto\$**, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- DATABASE\_PASSWORD
- DATABASE\_READ\_ONLY\_PASSWORD
- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD
- BDCS\_ADMIN\_PASSWORD
- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


- f) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the [OBFUSCATION\\_PREFIX](#) on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`;

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- g) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

```

• SSL=on
• SSL_CERT_FILE=
• SSL_KEY_FILE=
• SSL_CA_CERT_FILE=
• SSL_CHAIN_FILE=

```

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- h) If the token `REQUIRE_PASSWORD_SECURITY` is enabled, then set a value for the token, `PASSWORD_CONTROL_EFFECTIVE_DATE`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- i) 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.
- j) 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__}
```
- k) If you have LDAP set up for external authentication, you must set the `“REQUIRE_USER_PASSWORD_CHANGE”` site options token to false.
- l) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and Postgresql automatically.
- m) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

- n) Make sure that the following tokens have a value if ETL is enabled.

```
SOAP_ANONYMOUS_SHARED_SECRET=
ETL_SOAP_SHARED_SECRET=
```

- o) 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
```

- 👉 **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 username add this
token:
BDCS_ADMIN_USERNAME=<sysadmin>
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 are 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 the 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
```

- p) To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
  - q) If you are installing TeamForge through disconnected media, set the token **HELP\_AVAILABILITY=local**.
  - r) Ensure to set the token **SELINUX\_SETUP=false** temporarily in the `site-options.conf` file.
  - s) Save the `site-options.conf` file.
7. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

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

```
export http_proxy=
```



**Do this on the reporting server - my.reports.host**

9. Install Red Hat Enterprise Linux / CentOS 6.4 or later versions and log in as root.

- The host must be registered with the Red Hat Network if you are using Red Hat Enterprise Linux. See [Platform specification for TeamForge 7.1](#) on page 356 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. Check your basic networking setup.

See [Set up networking for your TeamForge server](#) on page 7 for details.

11. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.

c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```


12. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

13. Run the following command to install the Reporting packages.

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

14. Copy the `site-options.conf` file from the application server to the reporting server in the directory `/opt/collabnet/teamforge-installer/7.1.0.0/conf`

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

 **Note:** If you choose not to use the application server's `site-options.conf` file, then don't forget to copy the value of `AUTO_DATA` token from the application server.

```
HOST_localhost=datamart etl
```

```
DOMAIN_localhost=my.reports.domain.com
```

```
HOST_my.app.domain.com=app database indexer subversion cvs gerrit
codesearch
```

16. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```


**Do the following on the application server - my.app.host**

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

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./bootstrap-data.sh
```

18. Start TeamForge.

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

 **Note:** Startup can take several minutes, depending on the speed of the host system configuration. 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.

**Do this on the reporting server - my.reports.host**

19. Start the ETL service.

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

**Do the following on the application server - my.app.host**

20. If you have installed GIT, create the 'gitadmin' user (which is already added in the site-options token `USERS_WITH_NO_EXPIRY_PASSWORD`) with site administrator rights in the TeamForge user interface.

21. 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 302.

22. If you have installed Git, integrate gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

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

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

24. Run the following initial load jobs (ETL).

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 325.

25. 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.
- 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
```

26. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

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

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

28. 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
```

29. Restart the collabnet services.

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

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

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

- Create a sample project.

See [Create a TeamForge project](#).

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

See [Create a site-wide broadcast](#).

For specific instructions on installing Review Board, see: [Set up Review Board](#) on page 305

To run TeamForge in SELINUX enabled mode, see [Set up SELINUX](#) on page 271

## Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see [TeamForge Orchestrate installation](#).

### Install TeamForge 7.1 with Black Duck Code Sight on a separate server on Red Hat/CentOS

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



**Note:** For the ETL service to run as expected in a distributed TeamForge installation, all servers must have the same time zone.

In this option, the following services run on the application server (we call this `my.app.host`).

- TeamForge Application Server
- Database Server (Operational DB and Reports DB)
- ETL Server
- GIT Integration Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer).

The following service runs on the Black Duck Code Sight Server. (We call this `my.codesight.host`)

- Black Duck Code Sight Server

**Do this on the main TeamForge application server. We'll call this my.app.host.****1.** Install Red Hat Enterprise Linux / CentOS 6.4 or later versions and log in as root.

- The host must be registered with the Red Hat Network if you are using Red Hat Enterprise Linux. See [Platform specification for TeamForge 7.1](#) on page 356 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 server](#) on page 7 for details.

**3.** If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

## a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.

c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```

See [Set up SELINUX](#) on page 271 to have TeamForge to run in SELinux mode after completing the installation or upgrade.

**4.** Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)**5.** 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
```

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

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

a) Configure the HOST token.

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


```
DOMAIN_localhost=my.app.domain.com
```

```
HOST_my.codesight.domain.com=codesearch
```

b) Configure the following settings if you are installing Git.

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

c) Configure the database and datamart settings.

 **Note:** For more information about configuring variables, see [site-options.conf](#) on page 383

```
DATABASE_TYPE=postgresql
```

```
DATABASE_USERNAME=ctfuser
```

```
DATABASE_NAME=ctfdb
```

```
DATABASE_READ_ONLY_USER=ctfrouser
```

```
REPORTS_DATABASE_USERNAME=ctfrptuser
```

```
REPORTS_DATABASE_NAME=ctfrptdb
```

```
REPORTS_DATABASE_READ_ONLY_USER=ctfrptrouser
```

```
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

 **Note:** The database name and username values are arbitrary alphanumeric strings.

- d) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to **Sauto\$**, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- DATABASE\_PASSWORD
- DATABASE\_READ\_ONLY\_PASSWORD
- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD
- BDCS\_ADMIN\_PASSWORD
- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


- e) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the [OBFUSCATION\\_PREFIX](#) on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`;

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- f) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

```

• SSL=on
• SSL_CERT_FILE=
• SSL_KEY_FILE=
• SSL_CA_CERT_FILE=
• SSL_CHAIN_FILE=

```

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- g) If the token `REQUIRE_PASSWORD_SECURITY` is enabled, then set a value for the token, `PASSWORD_CONTROL_EFFECTIVE_DATE`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- h) 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.
- i) 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__}
```

- j) If you have LDAP set up for external authentication, you must set the `REQUIRE_USER_PASSWORD_CHANGE` site options token to false.
- k) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and Postgresql automatically.
- l) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

- 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.

 **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 username add this
token:
BDCS_ADMIN_USERNAME=<sysadmin>
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 are 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 the 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
```

- To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
  - If you are installing TeamForge through disconnected media, set the token **HELP\_AVAILABILITY=local**.
  - Ensure to set the token **SELINUX\_SETUP=false** temporarily in the `site-options.conf` file.
  - Save the `site-options.conf` file.
7. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

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

```
export http_proxy=
```

**Do this on the Black Duck Code Sight Server. We'll call this my.codesight.host**

9. Install Red Hat Enterprise Linux / CentOS 6.4 or later versions and log in as root.

- The host must be registered with the Red Hat Network if you are using Red Hat Enterprise Linux. See [Platform specification for TeamForge 7.1](#) on page 356 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. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.

c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```


11. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

12. Run the following command to install the Black Duck Code Sight packages.

```
yum install teamforge-codesearch
```

13. Copy the `site-options.conf` file from the application server to the Code Search server in the directory `/opt/collabnet/teamforge-installer/7.1.0.0/conf`

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

 **Note:** If you choose not to use the application server's `site-options.conf` file, then don't forget to copy the value of `AUTO_DATA` token from the application server.

```
HOST_localhost=codesearch
```

```
DOMAIN_localhost=my.codesight.domain.com
```

```
HOST_my.app.domain.com=app database datamart etl indexer subversion cvs
gerrit
```

15. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```


**Do the following on the application server - my.app.host**

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

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./bootstrap-data.sh
```

17. Start TeamForge.


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

 **Note:** Startup can take several minutes, depending on the speed of the host system configuration. 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.

18. If you have installed GIT, create the 'gitadmin' user (which is already added in the site-options token `USERS_WITH_NO_EXPIRY_PASSWORD`) with site administrator rights in the TeamForge user interface.

19. If you have installed Git, integrate gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

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
```

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

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

21. Run the following initial load jobs (ETL).

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 325.


22. Revoke the user permissions of the database and datamart users.

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

### Do this on my.codesight.host

23. Install the Black Duck Code Sight license on the server where Black Duck Code Sight is installed. For more information, see [Install the Black Duck Code Sight license](#) on page 302.

24. 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.

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

25. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

26. Restart the Black Duck Code Sight service.

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

### Do this on my.app.host

27. Restart the collabnet services.

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

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

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

b) 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.

c) Create a sample project.

See [Create a TeamForge project](#).

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

See [Create a site-wide broadcast](#).

For specific instructions on installing Review Board, see: [Set up Review Board](#) on page 305

To run TeamForge in SELINUX enabled mode, see [Set up SELINUX](#) on page 271

### Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see [TeamForge Orchestrate installation](#).

### Install TeamForge 7.1 with SCM and Git integration on a separate server


In this option, we install SCM (Subversion, CVS) and GIT integrations on a separate server and other services on the main application server.

In this option, the following services run on the application server (we call this my.app.host).

- TeamForge Application Server
- Black Duck Code Sight Server
- Database Server (Operational DB and Reports DB)
- ETL Server
- Search Server (Indexer).

The following service runs on the SCM server. (We call this my.scmandgit.host)

- SCM Integration Server (Subversion and CVS)
- GIT Integration Server


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

Log in to the server as root.

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

1. Install Red Hat Enterprise Linux / CentOS 6.4 or later versions and log in as root.

- The host must be registered with the Red Hat Network if you are using Red Hat Enterprise Linux. See [Platform specification for TeamForge 7.1](#) on page 356 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 server](#) on page 7 for details.

3. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

- a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

- b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.

- c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```

See [Set up SELINUX](#) on page 271 to have TeamForge to run in SELinux mode after completing the installation or upgrade.

4. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

5. Install the following application packages.

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

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

- 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/7.1.0.0/conf/site-options.conf
```

- a) Identify the servers and services running on them.

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


```
DOMAIN_localhost=my.app.domain.com
```

```
HOST_<my.scmangit.domain.com>=subversion cvs Gerrit
```

- b) Configure the following settings if you are installing Black Duck Code Sight.

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

- c) Configure the database and datamart settings.

 **Note:** For more information about configuring variables, see [site-options.conf](#) on page 383

```
DATABASE_TYPE=postgresql
```

```
DATABASE_USERNAME=ctfuser
```

```
DATABASE_NAME=ctfdb
```


```
DATABASE_READ_ONLY_USER=ctfrouser
```

```
REPORTS_DATABASE_USERNAME=ctfrptuser
```

```
REPORTS_DATABASE_NAME=ctfrptdb
```

```
REPORTS_DATABASE_READ_ONLY_USER=ctfrptrouser
```

```
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

 **Note:** The database name and username values are arbitrary alphanumeric strings.

- d) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to **Sauto\$**, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- DATABASE\_PASSWORD
- DATABASE\_READ\_ONLY\_PASSWORD
- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD
- BDCS\_ADMIN\_PASSWORD
- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


- e) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`;

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- f) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

```

• SSL=on
• SSL_CERT_FILE=
• SSL_KEY_FILE=
• SSL_CA_CERT_FILE=
• SSL_CHAIN_FILE=

```

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- g) If the token `REQUIRE_PASSWORD_SECURITY` is enabled, then set a value for the token, `PASSWORD_CONTROL_EFFECTIVE_DATE`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- h) 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.
- i) 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__}
```

- j) If you have LDAP set up for external authentication, you must set the `“REQUIRE_USER_PASSWORD_CHANGE”` site options token to false.
- k) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and Postgresql automatically.
- l) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

- 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.

 **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 username add this
token:
BDCS_ADMIN_USERNAME=<sysadmin>
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 are 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 the 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
```

- To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
  - If you are installing TeamForge through disconnected media, set the token **HELP\_AVAILABILITY=local**.
  - Ensure to set the token **SELINUX\_SETUP=false** temporarily in the `site-options.conf` file.
  - Save the `site-options.conf` file.
7. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

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

```
export http_proxy=
```

**Do this on the SCM Server - myscmandgit.host**

9. Install Red Hat Enterprise Linux / CentOS 6.4 or later versions and log in as root.

- The host must be registered with the Red Hat Network if you are using Red Hat Enterprise Linux. See [Platform specification for TeamForge 7.1](#) on page 356 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. Check your basic networking setup.

See [Set up networking for your TeamForge server](#) on page 7 for details.

11. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.

c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```


12. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

13. Install the TeamForge SCM and Git packages.

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

14. Copy the `site-options.conf` file from the application server to the SCM server in the directory `/opt/collabnet/teamforge-installer/7.1.0.0/conf`

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

 **Note:** If you choose not to use the application server's `site-options.conf` file, then don't forget to copy the value of `AUTO_DATA` token from the application server.

```
HOST_localhost=subversion cvs gerrit
```

```
DOMAIN_localhost=my.scmmandgit.domain.com
```

```
HOST_my.app.domain.com=app database datamart etl indexer codesearch
```

16. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

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

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./bootstrap-data.sh
```


**Do the following on the application server - my.app.host**

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

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./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 configuration. 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. 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 302.

21. Run the following initial load jobs (ETL).

- 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 325.

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

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


23. 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.
- 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
```

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

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

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL path of the SCM integration server, for example, "http://myint.box.net/svn/repos", where `myint.box` is the server with the SCM integration server.

In addition, if you add a new integration server at some point later, you must run this `svn_cache.sh` script, (after creating the new integration server), on the TeamForge application server.



25. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

#### Do this on the SCM server - myscmandgit.host

26. If you have installed GIT, create the 'gitadmin' user (which is already added in the site-options token `USERS_WITH_NO_EXPIRY_PASSWORD`) with site administrator rights in the TeamForge user interface.

27. If you have installed Git, integrate gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

- 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
```

#### Do the following on the application server - my.app.host

28. Restart the collabnet services.

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

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

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

- b) 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.

- c) Create a sample project.

See [Create a TeamForge project](#).

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

See [Create a site-wide broadcast](#).

For specific instructions on installing Review Board, see: [Set up Review Board](#) on page 305

To run TeamForge in SELINUX enabled mode, see [Set up SELINUX](#) on page 271

### Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see [TeamForge Orchestrate installation](#).

#### Install TeamForge 7.1 with Database and SCM on separate servers

In this option, we install the Database (Operational Database) and Datamart (Reporting Database) on the same server; SCM (Subversion and CVS) and Git on the second server, and other services on the application server.

In this option, the following services run on the application server (we call this my.app.host).

- TeamForge Application Server
- Black Duck Code Sight Server
- ETL Server


- Search Server (Indexer).

The following service runs on the database server. (We call this my.db.host)

- Database Server (Operational DB and Reports DB)

The following services run on the SCM server. (We call this myscmandgit.host)

- SCM Integration Server (Subversion and CVS)
- GIT Integration Server

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

Log in to the server as root.

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

1. Install Red Hat Enterprise Linux / CentOS 6.4 or later versions and log in as root.

- The host must be registered with the Red Hat Network if you are using Red Hat Enterprise Linux. See [Platform specification for TeamForge 7.1](#) on page 356 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 server](#) on page 7 for details.

3. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

- a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

- b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.

- c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```

See [Set up SELINUX](#) on page 271 to have TeamForge to run in SELinux mode after completing the installation or upgrade.

4. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

5. Install the following application packages.

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

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

- 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/7.1.0.0/conf/site-options.conf
```

- a) Identify the servers and services running on them.

```
HOST_localhost=app etl indexer
```

```
DOMAIN_localhost=my.app.domain.com
```


```
HOST_<my.db.domain.com>=database datamart
```

```
HOST_<my.scmangit.domain.com>=subversion cvs gerrit
```

- b) Configure the following settings if you are installing Black Duck Code Sight.

```
HOST_localhost=app etl indexer codesearch
```

- c) Configure the database and datamart settings.

 **Note:** For more information about configuring variables, see [site-options.conf](#) on page 383

```
DATABASE_TYPE=postgresql
```

```
DATABASE_USERNAME=ctfuser
```

```
DATABASE_NAME=ctfdb
```


```
DATABASE_READ_ONLY_USER=ctfrouser
```

```
REPORTS_DATABASE_USERNAME=ctfrptuser
```

```
REPORTS_DATABASE_NAME=ctfrptdb
```

```
REPORTS_DATABASE_READ_ONLY_USER=ctfrptrouser
```

```
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

 **Note:** The database name and username values are arbitrary alphanumeric strings.

- d) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to **Sauto\$**, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- DATABASE\_PASSWORD
- DATABASE\_READ\_ONLY\_PASSWORD
- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD
- BDCS\_ADMIN\_PASSWORD
- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


- e) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any Alphanumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the [OBFUSCATION\\_PREFIX](#) on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`).

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- f) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

```

• SSL=on
• SSL_CERT_FILE=
• SSL_KEY_FILE=
• SSL_CA_CERT_FILE=
• SSL_CHAIN_FILE=

```

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- g) If the token `REQUIRE_PASSWORD_SECURITY` is enabled, then set a value for the token, `PASSWORD_CONTROL_EFFECTIVE_DATE`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- h) 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.
- i) 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__}
```

- j) If you have LDAP set up for external authentication, you must set the `“REQUIRE_USER_PASSWORD_CHANGE”` site options token to false.
- k) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and Postgresql automatically.
- l) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

- 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.

 **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 username add this
token:
BDCS_ADMIN_USERNAME=<sysadmin>
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 are 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 the 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
```

- To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
  - If you are installing TeamForge through disconnected media, set the token **HELP\_AVAILABILITY=local**.
  - Ensure to set the token **SELINUX\_SETUP=false** temporarily in the `site-options.conf` file.
  - Save the `site-options.conf` file.
7. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```


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

```
export http_proxy=
```

**Do this on the database server - my.db.host**

9. Install Red Hat Enterprise Linux / CentOS 6.4 or later versions and log in as root.

- The host must be registered with the Red Hat Network if you are using Red Hat Enterprise Linux. See [Platform specification for TeamForge 7.1](#) on page 356 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. Check your basic networking setup.

See [Set up networking for your TeamForge server](#) on page 7 for details.

11. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.

c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```


12. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

13. Install the TeamForge database packages.

```
yum install teamforge-database
```

14. Copy the `site-options.conf` file from the application server to the database server in the directory `/opt/collabnet/teamforge-installer/7.1.0.0/conf`

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

 **Note:** If you choose not to use the application server's `site-options.conf` file, then don't forget to copy the value of `AUTO_DATA` token from the application server.

```
HOST_localhost=database datamart
```

```
DOMAIN_localhost=my.db.domain.com
```

```
HOST_my.app.domain.com=app etl indexer codesearch
```

```
HOST_<my.scmangit.domain.com>=subversion cvs gerrit
```

16. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

**Do this on the SCM Server - myscmandgit.host**

17. Install Red Hat Enterprise Linux / CentOS 6.4 or later versions and log in as root.

- The host must be registered with the Red Hat Network if you are using Red Hat Enterprise Linux. See [Platform specification for TeamForge 7.1](#) on page 356 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 server](#) on page 7 for details.

19. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

- b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.
- c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```


20. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

21. Install the TeamForge SCM and Git packages.

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

22. Copy the `site-options.conf` file from the application server to the SCM server in the directory `/opt/collabnet/teamforge-installer/7.1.0.0/conf`

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

-  **Note:** If you choose not to use the application server's `site-options.conf` file, then don't forget to copy the value of `AUTO_DATA` token from the application server.

```
HOST_localhost=subversion cvs gerrit
```

```
DOMAIN_localhost=my.scmangit.domain.com
```

```
HOST_my.app.domain.com=app etl indexer codesearch
```

```
HOST_<my.db.domain.com>=database datamart
```

24. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

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

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./bootstrap-data.sh
```


**Do the following on the application server - my.app.host**

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

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./bootstrap-data.sh
```


27. Start TeamForge.

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

-  **Note:** Startup can take several minutes, depending on the speed of the host system configuration. 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.

28. 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 302.

29. Run the following initial load jobs (ETL).

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 325.

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

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


31. 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.
- 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
```

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

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

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL path of the SCM integration server, for example, "`http://myint.box.net/svn/repos`", where `myint.box` is the server with the SCM integration server.

In addition, if you add a new integration server at some point later, you must run this `svn_cache.sh` script, (after creating the new integration server), on the TeamForge application server.

33. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

**Do this on the SCM server - myscmandgit.host**

34. If you have installed GIT, create the 'gitadmin' user (which is already added in the site-options token `USERS_WITH_NO_EXPIRY_PASSWORD`) with site administrator rights in the TeamForge user interface.

35. If you have installed Git, integrate gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.



- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

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
```

### Do the following on the application server - my.app.host

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

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

37. 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
```

38. 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
```

39. Restart the collabnet services.

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

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

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

b) 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.

c) Create a sample project.

See [Create a TeamForge project](#).

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

See [Create a site-wide broadcast](#).

For specific instructions on installing Review Board, see: [Set up Review Board](#) on page 305

To run TeamForge in SELINUX enabled mode, see [Set up SELINUX](#) on page 271

### Installing TeamForge Orchestrate


To install TeamForge Orchestrate, see [TeamForge Orchestrate installation](#).

#### Install Git integration on a separate server

In this option, we install the GIT integration services on a separate server.

In this option, the following service runs on the Git server (we call this my.git.host).

- GIT Integration Server

 **Note:** For the ETL service to run as expected in a distributed TeamForge installation, all servers must have the same time zone.

Log in to the server as root.

**Do this on the Git server. We'll call this my.git.host.**

1. Install Red Hat Enterprise Linux / CentOS 6.4 or later versions and log in as root.
  - The host must be registered with the Red Hat Network if you are using Red Hat Enterprise Linux. See [Platform specification for TeamForge 7.1](#) on page 356 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 server](#) on page 7 for details.

3. If the Git server has SELinux enabled, disable it temporarily while installing or upgrading Git.

- a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

- b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.

- c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```

4. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)


5. Install the Git packages.

```
yum install teamforge-git
```

6. Configure the token settings for Git in the `site-options.conf` file.

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

- a) Modify the host token settings.

 **Note:** If you choose not to use the application server's `site-options.conf` file, then don't forget to copy the value of `AUTO_DATA` token from the application server.

```
HOST_localhost=gerrit
```

```
DOMAIN_localhost=my.git.domain.com
```

```
HOST_my.app.domain.com=app database datamart etl indexer subversion cvs
```

- b) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and Postgresql automatically.

- c) To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396

- d) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

- ```
SSL=on
```

- ```
SSL_CERT_FILE=
```

- ```
SSL_KEY_FILE=
```

- ```
SSL_CA_CERT_FILE=
```

- ```
SSL_CHAIN_FILE=
```

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

e) Save the `site-options.conf` file.

7. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

8. Create the 'gitadmin' user (which is already added in the `site-options` token `USERS_WITH_NO_EXPIRY_PASSWORD`) with site administrator rights in the TeamForge user interface.

9. If you have installed Git, integrate gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

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](#).

### 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 server or on different servers, in whatever combination works best for your conditions.

In principle, a multi-server 7.1 site can have its services running in a wide variety of combinations on an undefined number of servers. However, real-world sites tend to follow one of the following patterns, depending on the specific needs of the community of site users.

### Install TeamForge 7.1 with Oracle database on a separate server


In this option, we install the Oracle database (Operational database and Reports database) on a separate server and other services on the main application server.

In this option, the following services run on the application server (we call this `my.app.host`).

- TeamForge Application Server
- Black Duck Code Sight Server
- ETL Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer).


The following service runs on the database server. (We call this `my.db.host`)

- Database Server (Operational DB and Reports DB)

-  **Note:** If either of the remote servers (the data server or the source code server) is not under your direct control, check with the Database Administrator to make sure that you can carry out these instructions on that server.

1. Install Red Hat Enterprise Linux / CentOS 6.4 or later versions and log in as root.

- The host must be registered with the Red Hat Network if you are using Red Hat Enterprise Linux. See [Platform specification for TeamForge 7.1](#) on page 356 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 server](#) on page 7 for details.

3. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

- a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

- b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.

- c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```

See [Set up SELINUX](#) on page 271 to have TeamForge to run in SELinux mode after completing the installation or upgrade.

4. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

5. Install the following application packages.

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


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

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

```
yum install teamforge-codesearch
```

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

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
cp conf/site-options-advanced.conf conf/site-options.conf
```

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

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

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

- a) Configure the HOST token.

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


```
DOMAIN_localhost=my.app.domain.com
```

```
HOST_<my.db.host>=database datamart
```

- b) Configure the following settings if you are installing Black Duck Code Sight.

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

- c) Configure the database and datamart settings.

 **Note:** For more information about configuring variables, see [site-options.conf](#) on page 383

```
DATABASE_TYPE=oracle
```

```
DATABASE_USERNAME=sitedatabaseusername
```

```
DATABASE_PASSWORD=sitedatabasepwd
```

```
DATABASE_READ_ONLY_USER=sitedatabasereadonlyusername
```

```
DATABASE_READ_ONLY_PASSWORD=sitedatabasereadonlyuserpwd
```

```
DATABASE_NAME=sitedatabaseinstancename
```

```
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
```

```
DATABASE_SERVICE_NAME=
```

```
REPORTS_DATABASE_SERVICE_NAME=
```

 **Tip:** To find the value for the token `DATABASE_SERVICE_NAME` log in to your Oracle server and execute this command.

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

Find the value of the `SERVICE_NAME` in the output and use this value for the `DATABASE_SERVICE_NAME` in the `site-options.conf` file.


#### d) Password Obfuscation

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the [OBFUSCATION\\_PREFIX](#) on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`;

 **Important:** The password-related tokens cannot contain the following characters: `$<>\/\ ' " `` in the `site-options.conf` file.

- e) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

```

• SSL=on
• SSL_CERT_FILE=
• SSL_KEY_FILE=
• SSL_CA_CERT_FILE=
• SSL_CHAIN_FILE=

```

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- f) If the token `REQUIRE_PASSWORD_SECURITY` is enabled, then set a value for the token, `PASSWORD_CONTROL_EFFECTIVE_DATE`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- g) 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.
- h) If the token `REQUIRE_RANDOM_ADMIN_PASSWORD` is already set to true, then set the token `ADMIN_EMAIL` with a valid email address.
- i) If you have LDAP set up for external authentication, you must set the `REQUIRE_USER_PASSWORD_CHANGE` site options token to false.
- j) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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) 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.

 **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 username add this
token:
BDCS_ADMIN_USERNAME=<sysadmin>
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 are 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 the 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
```

- m) If you are installing TeamForge through disconnected media, set the token *HELP\_AVAILABILITY=local*.
- n) Ensure to set the token, *SELINUX\_SETUP=false* temporarily in the *site-options.conf* file.
- o) Save the *site-options.conf* file.

#### 8. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

#### 9. If you are installing on a server that is behind a proxy, unset the *http\_proxy* token.

```
export http_proxy=
```

#### 10. 👉 **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>
```

**11. Recreate the runtime environment.**

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

**12. Copy the Oracle datamart setup script from /opt/collabnet/teamforge/runtime/scripts to the /tmp directory of my.db.host.**

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


**Do this on the database server my.db.host**

**13. Install Red Hat Enterprise Linux / CentOS 6.4 or later versions and log in as root.**

- The host must be registered with the Red Hat Network if you are using Red Hat Enterprise Linux. See [Platform specification for TeamForge 7.1](#) on page 356 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 Oracle 11G (R1 and R2).**

 **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](#).


**15. Copy the Oracle datamart setup script.**

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

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

See [Set up an Oracle database](#) on page 263 for help.

**17. Create the reporting user and schema.**

 **Note:** Skip this step if you have already set up the datamart setup in the Oracle database. Your responses to the script's prompts must match the values of the equivalent variables in the `site-options.conf` file on `my.app.server`.

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

**Do this on the TeamForge Application Server (my.app.host)**

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

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

**19. 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
```

**20. 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
```

**21. 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
```

## 22. 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.

## 23. 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 302.

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

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

## 25. Run the following initial load jobs (ETL).

### 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 325.

## 26. 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.

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

## 27. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

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

```
./trust-cert.sh
```

```
/etc/init.d/collabnet -V restart jboss
```

## 28. Restart the collabnet services.

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

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

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

### b) 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.

### c) Create a sample project.

See [Create a TeamForge project](#).

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

See [Create a site-wide broadcast](#).

For specific instructions on installing Review Board, see: [Set up Review Board](#) on page 305

To run TeamForge in SELINUX enabled mode, see [Set up SELINUX](#) on page 271


To install Git integration see [Install Git integration on a separate server](#) on page 49

### Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see [TeamForge Orchestrate installation](#).

### Uninstall TeamForge 7.1

To remove TeamForge completely, use the YUM utility.

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

1. Stop TeamForge.

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

2. Run yum to remove TeamForge.

```
yum erase TeamForge-installer
```

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


## Install CollabNet TeamForge 7.1 on SuSE

### Install TeamForge 7.1 with all services on the same server

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

In this option, the following services run on the application server (we call this my.app.host).

- TeamForge Application Server
- Black Duck Code Sight Server
- Database Server (Operational DB and Reports DB)
- ETL Server
- GIT Integration Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer).

 **Important:** It is *critical* that you start with a fresh server, without any software installed. 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 7.1](#) on page 356 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 server](#) on page 7 for details.

3. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

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

```
zypper install teamforge-codesearch
```

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

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

a) Configure the HOST token.

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

```
DOMAIN_localhost=my.app.domain.com
```


b) Configure the following settings if you are installing Git.

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

c) Configure the following settings if you are installing Black Duck Code Sight.

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

d) Configure the database and datamart settings.

 **Note:** For more information about configuring variables, see [site-options.conf](#) on page 383

```
DATABASE_TYPE=postgresql
```

```
DATABASE_USERNAME=ctfuser
```

```
DATABASE_NAME=ctfdb
```


```
DATABASE_READ_ONLY_USER=ctfrouser
```

```
REPORTS_DATABASE_USERNAME=ctfrptuser
```

```
REPORTS_DATABASE_NAME=ctfrptdb
```

```
REPORTS_DATABASE_READ_ONLY_USER=ctfrptrouser
```

```
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

 **Note:** The database name and username values are arbitrary alphanumeric strings.

e) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to **\$auto\$**, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- DATABASE\_PASSWORD
- DATABASE\_READ\_ONLY\_PASSWORD
- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD
- BDCS\_ADMIN\_PASSWORD

- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


#### f) Password Obfuscation

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSSION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSSION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSSION_KEY` token is `XSJt43wN`.

To configure the `OBFUSSION_PREFIX` on page 405, set `OBFUSSION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSSION_PREFIX` is `{OBF};`.

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- g) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

- `SSL=on`
- `SSL_CERT_FILE=`
- `SSL_KEY_FILE=`
- `SSL_CA_CERT_FILE=`
- `SSL_CHAIN_FILE=`

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- h) If the token `REQUIRE_PASSWORD_SECURITY` is enabled, then set a value for the token, `PASSWORD_CONTROL_EFFECTIVE_DATE`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- i) 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.
- j) 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__}
```
- k) If you have LDAP set up for external authentication, you must set the `“REQUIRE_USER_PASSWORD_CHANGE”` site options token to false.
- l) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and PostgreSQL automatically.
- m) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

- n) Make sure that the following tokens have a value if ETL is enabled.


```
SOAP_ANONYMOUS_SHARED_SECRET=
ETL_SOAP_SHARED_SECRET=
```

- o) 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

-  **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 username add this token:

```
BDCS_ADMIN_USERNAME=<sysadmin>
```

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 are 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 the 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

- p) To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
- q) If you are installing TeamForge through disconnected media, set the token `HELP_AVAILABILITY=local`.
- r) Save the `site-options.conf` file.

6. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

7. If you are installing on a server that is behind a proxy, unset the `http_proxy` token.

```
export http_proxy=
```

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 Git, add the "gitadmin" user with site-administrator rights through TeamForge user interface.

11. 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 302.

12. If you have installed Git, integrate gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

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
```

13. Run the following initial load jobs (ETL).

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 325.

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

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

15. 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.

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

16. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

17. Restart the collabnet services.

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

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

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

- c) Create a sample project.  
See [Create a TeamForge project](#).
- d) Write a welcome message to your site's users.  
See [Create a site-wide broadcast](#).

For specific instructions on installing Review Board, see: [Set up Review Board](#) on page 305

### Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see [TeamForge Orchestrate installation](#).

### Install TeamForge 7.1 with Database and Datamart on a separate server

In this option, we install the database (Operational Database) and datamart (Reporting database) on a separate server and other services on the main application server.


In this option, the following services run on the application server (we call this `my.app.host`).

- TeamForge Application Server
- Black Duck Code Sight Server
- ETL Server
- GIT Integration Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer).

The following service runs on the database server. (We call this `my.db.host`)

- Database Server (Operational DB and Reports DB)


We call this a "dedicated install" since both the servers are dedicated to TeamForge. The TeamForge 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.

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

**Log in to the server as root.**

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

1. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.
  - See [Platform specification for TeamForge 7.1](#) on page 356 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 server](#) on page 7 for details.
3. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)
4. Install the following application packages.

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

```
zypper install teamforge-app teamforge-etl teamforge-scm
```

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

```
zypper install teamforge-git
```

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

```
zypper install teamforge-codesearch
```

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

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

- a) Identify the servers and services running on them.

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

```
DOMAIN_localhost=my.app.domain.com
```

```
HOST_<my.db.domain.com>=database datamart
```

- b) Configure the following settings if you are installing Git.


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

- c) Configure the following settings if you are installing Black Duck Code Sight.

```
HOST_localhost=app etl indexer subversion cvs gerrit codesearch
```

- d) Configure the database and datamart settings.



 **Note:** For more information about configuring variables, see [site-options.conf](#) on page 383

```
DATABASE_TYPE=postgresql
```

```
DATABASE_USERNAME=ctfuser
```

```
DATABASE_NAME=ctfdb
```


```
DATABASE_READ_ONLY_USER=ctfrouser
```

```
REPORTS_DATABASE_USERNAME=ctfrptuser
```

```
REPORTS_DATABASE_NAME=ctfrptdb
```

```
REPORTS_DATABASE_READ_ONLY_USER=ctfrptrouser
```

```
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

 **Note:** The database name and username values are arbitrary alphanumeric strings.

- e) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to **\$auto\$**, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- DATABASE\_PASSWORD
- DATABASE\_READ\_ONLY\_PASSWORD
- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD
- BDCS\_ADMIN\_PASSWORD
- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


- f) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`;

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- g) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

```

• SSL=on
• SSL_CERT_FILE=
• SSL_KEY_FILE=
• SSL_CA_CERT_FILE=
• SSL_CHAIN_FILE=

```

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- h) If the token `REQUIRE_PASSWORD_SECURITY` is enabled, then set a value for the token, `PASSWORD_CONTROL_EFFECTIVE_DATE`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- i) 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.
- j) 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__}
```
- k) If you have LDAP set up for external authentication, you must set the `REQUIRE_USER_PASSWORD_CHANGE` site options token to false.
- l) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and Postgresql automatically.
- m) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

- n) Make sure that the following tokens have a value if ETL is enabled.

```
SOAP_ANONYMOUS_SHARED_SECRET=
ETL_SOAP_SHARED_SECRET=
```

- o) 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
```

- 👉 **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 username add this
token:
BDCS_ADMIN_USERNAME=<sysadmin>
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 are 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 the 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
```

- To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
- If you are installing TeamForge through disconnected media, set the token **HELP\_AVAILABILITY=local**.
- Save the `site-options.conf` file.

#### 6. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

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

```
export http_proxy=
```

#### Do this on the database server - my.db.host

8. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See [Platform specification for TeamForge 7.1](#) on page 356 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.

9. Check your basic networking setup.

See [Set up networking for your TeamForge server](#) on page 7 for details.


10. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

11. Install the TeamForge database packages.

```
zypper install teamforge-database
```

12. Copy the `site-options.conf` file from the application server to the database server in the directory `/opt/collabnet/teamforge-installer/7.1.0.0/conf`

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

 **Note:** If you choose not to use the application server's `site-options.conf` file, then don't forget to copy the value of `AUTO_DATA` token from the application server.

```
HOST_localhost=database datamart
```

```
DOMAIN_localhost=my.db.domain.com
```

```
HOST_my.app.domain.com=app etl indexer subversion cvs gerrit codesearch
```

14. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

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 to set permissions for the reporting database read-only user.

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


**Do the following on the application server - my.app.host**

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

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./bootstrap-data.sh
```

18. Start TeamForge.

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

 **Note:** Startup can take several minutes, depending on the speed of the host system configuration. 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.

19. If you have installed GIT, create the 'gitadmin' user (which is already added in the site-options token `USERS_WITH_NO_EXPIRY_PASSWORD`) with site administrator rights in the TeamForge user interface.

20. 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 302.

21. If you have installed Git, integrate gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

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
```

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

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

23. Run the following initial load jobs (ETL).

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 325.

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.
- 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. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

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

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

27. 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
```

28. 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
```

29. Restart the collabnet services.

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

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

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

b) 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.

c) Create a sample project.

See [Create a TeamForge project](#).

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

See [Create a site-wide broadcast](#).


For specific instructions on installing Review Board, see: [Set up Review Board](#) on page 305

### Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see [TeamForge Orchestrate installation](#).

### Install TeamForge 7.1 with Reporting services on a separate server

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

 **Note:** For the ETL service to run as expected in a distributed TeamForge installation, all servers must have the same time zone.

In this option, the following services run on the application server (we call this `my.app.host`).

- TeamForge Application Server
- Black Duck Code Sight Server
- Database Server (Operational DB)
- GIT Integration Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer).


The following service runs on the database server. (We call this `my.reports.host`)

- Database Server (Reports DB)
- ETL Server

Log in to the server as root.

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

1. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.
  - See [Platform specification for TeamForge 7.1](#) on page 356 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 server](#) on page 7 for details.
3. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)
4. Install the following application packages.
  - a) TeamForge: To install the TeamForge application packages run the following command:
 

```
zypper install teamforge-app teamforge-database teamforge-scm
```
  - b) GIT: To install the GIT packages run the following command.
 

```
zypper install teamforge-git
```
  - c) To install Black Duck Code Sight run the following command.
 

```
zypper install teamforge-codesearch
```
5. Set up your site's master configuration file.
 

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

  - a) Identify the servers and services running on them.
 


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

```
DOMAIN_localhost=my.app.domain.com
```

```
HOST_<my.reports.domain.com>=datamart etl
```
  - b) Configure the following settings if you are installing Git.
 

```
HOST_localhost=app database indexer subversion cvs Gerrit
```
  - c) Configure the following settings if you are installing Black Duck Code Sight.
 

```
HOST_localhost=app database indexer subversion cvs Gerrit codesearch
```
  - d) Configure the database and datamart settings.

 **Note:** For more information about configuring variables, see [site-options.conf](#) on page 383

```
DATABASE_TYPE=postgresql
```

```
DATABASE_USERNAME=ctfuser
```

```
DATABASE_NAME=ctfdb
```


```
DATABASE_READ_ONLY_USER=ctfrouser
```

```
REPORTS_DATABASE_USERNAME=ctfrptuser
```

```
REPORTS_DATABASE_NAME=ctfrptdb
```

```
REPORTS_DATABASE_READ_ONLY_USER=ctfrptrouser
```

```
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

 **Note:** The database name and username values are arbitrary alphanumeric strings.

- e) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to **Sauto\$**, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- DATABASE\_PASSWORD
- DATABASE\_READ\_ONLY\_PASSWORD
- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD
- BDCS\_ADMIN\_PASSWORD
- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


- f) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the [OBFUSCATION\\_PREFIX](#) on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF};`.

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.



- g) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

```

• SSL=on
• SSL_CERT_FILE=
• SSL_KEY_FILE=
• SSL_CA_CERT_FILE=
• SSL_CHAIN_FILE=

```

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- h) If the token `REQUIRE_PASSWORD_SECURITY` is enabled, then set a value for the token, `PASSWORD_CONTROL_EFFECTIVE_DATE`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- i) 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.
- j) 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__}
```
- k) If you have LDAP set up for external authentication, you must set the `"REQUIRE_USER_PASSWORD_CHANGE"` site options token to false.
- l) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and Postgresql automatically.
- m) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

- n) Make sure that the following tokens have a value if ETL is enabled.

```
SOAP_ANONYMOUS_SHARED_SECRET=
ETL_SOAP_SHARED_SECRET=
```

- o) 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
```

- 👉 **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 username add this
token:
BDCS_ADMIN_USERNAME=<sysadmin>
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 are 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 the 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
```

- To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
- If you are installing TeamForge through disconnected media, set the token **HELP\_AVAILABILITY=local**.
- Save the `site-options.conf` file.

#### 6. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

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

```
export http_proxy=
```

**Do this on the reporting server - my.reports.host**

8. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See [Platform specification for TeamForge 7.1](#) on page 356 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.

9. Check your basic networking setup.

See [Set up networking for your TeamForge server](#) on page 7 for details.


10. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

11. Run the following command to install the Reporting packages.

```
zypper install teamforge-database teamforge-etl
```

12. Copy the `site-options.conf` file from the application server to the reporting server in the directory `/opt/collabnet/teamforge-installer/7.1.0.0/conf`

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

 **Note:** If you choose not to use the application server's `site-options.conf` file, then don't forget to copy the value of `AUTO_DATA` token from the application server.

```
HOST_localhost=datamart etl
```

```
DOMAIN_localhost=my.reports.domain.com
```

```
HOST_my.app.domain.com=app database indexer subversion cvs gerrit
codesearch
```

14. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```


**Do the following on the application server - my.app.host**

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

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./bootstrap-data.sh
```

16. Start TeamForge.

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

 **Note:** Startup can take several minutes, depending on the speed of the host system configuration. 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.

**Do this on the reporting server - my.reports.host**

17. Start the ETL service.

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

**Do the following on the application server - my.app.host**

18. If you have installed GIT, create the 'gitadmin' user (which is already added in the site-options token `USERS_WITH_NO_EXPIRY_PASSWORD`) with site administrator rights in the TeamForge user interface.

19. 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 302.

20. If you have installed Git, integrate gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

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
```

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

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

22. Run the following initial load jobs (ETL).

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 325.

23. 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.

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

24. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

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

```
./trust-cert.sh
```

```
/etc/init.d/collabnet -V restart jboss
```

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
```

27. Restart the collabnet services.

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

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

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

b) 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.

c) Create a sample project.

See [Create a TeamForge project](#).

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

See [Create a site-wide broadcast](#).


For specific instructions on installing Review Board, see: [Set up Review Board](#) on page 305

### Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see [TeamForge Orchestrate installation](#).

### Install TeamForge 7.1 with Black Duck Code Sight on a separate server on SUSE

In this option, we install Black Duck Code Sight on a separate server on SUSE and other services on the main application server.

 **Note:** For the ETL service to run as expected in a distributed TeamForge installation, all servers must have the same time zone.

In this option, the following services run on the application server (we call this `my.app.host`).

- TeamForge Application Server
- Database Server (Operational DB and Reports DB)
- ETL Server
- GIT Integration Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer).

The following service runs on the Black Duck Code Sight Server. (We call this `my.codesight.host`)

- Black Duck Code Sight Server

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

1. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See [Platform specification for TeamForge 7.1](#) on page 356 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 server](#) on page 7 for details.

3. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

4. 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
```

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

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

- a) Configure the HOST token.

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


```
DOMAIN_localhost=my.app.domain.com
```

```
HOST_my.codesight.domain.com=codesearch
```

- b) Configure the following settings if you are installing Git.

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

- c) Configure the database and datamart settings.

 **Note:** For more information about configuring variables, see [site-options.conf](#) on page 383

```
DATABASE_TYPE=postgresql
```

```
DATABASE_USERNAME=ctfuser
```

```
DATABASE_NAME=ctfdb
```


```
DATABASE_READ_ONLY_USER=ctfrouser
```

```
REPORTS_DATABASE_USERNAME=ctfrptuser
```

```
REPORTS_DATABASE_NAME=ctfrptdb
```

```
REPORTS_DATABASE_READ_ONLY_USER=ctfrptrouser
```

```
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

 **Note:** The database name and username values are arbitrary alphanumeric strings.

- d) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to **\$auto\$**, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- DATABASE\_PASSWORD
- DATABASE\_READ\_ONLY\_PASSWORD
- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD

- BDCS\_ADMIN\_PASSWORD
- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


#### e) Password Obfuscation

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`:

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- f) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

- `SSL=on`
- `SSL_CERT_FILE=`
- `SSL_KEY_FILE=`
- `SSL_CA_CERT_FILE=`
- `SSL_CHAIN_FILE=`

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- g) If the token `REQUIRE_PASSWORD_SECURITY` is enabled, then set a value for the token, `PASSWORD_CONTROL_EFFECTIVE_DATE`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- h) 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.
- i) 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__}
```
- j) If you have LDAP set up for external authentication, you must set the `"REQUIRE_USER_PASSWORD_CHANGE"` site options token to false.
- k) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and Postgresql automatically.
- l) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

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.

 **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 username add this token:  
BDCS\_ADMIN\_USERNAME=<sysadmin>  
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 are 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 the 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



- o) To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
- p) If you are installing TeamForge through disconnected media, set the token `HELP_AVAILABILITY=local`.
- q) Save the `site-options.conf` file.

6. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

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

```
export http_proxy=
```

**Do this on the Black Duck Code Sight Server. We'll call this `my.codesight.host`**

8. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See [Platform specification for TeamForge 7.1](#) on page 356 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.

9. Check your basic networking setup.

See [Set up networking for your TeamForge server](#) on page 7 for details.


10. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

11. Run the following command to install the Black Duck Code Sight packages.

```
zypper install teamforge-codesearch
```

12. Copy the `site-options.conf` file from the application server to the Code Search server in the directory `/opt/collabnet/teamforge-installer/7.1.0.0/conf`

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

 **Note:** If you choose not to use the application server's `site-options.conf` file, then don't forget to copy the value of `AUTO_DATA` token from the application server.

```
HOST_localhost=codesearch
```

```
DOMAIN_localhost=my.codesight.domain.com
```

```
HOST_my.app.domain.com=app database datamart etl indexer subversion cvs
gerrit
```

14. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```


**Do the following on the application server - `my.app.host`**

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

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./bootstrap-data.sh
```

16. Start TeamForge.

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

-  **Note:** Startup can take several minutes, depending on the speed of the host system configuration. 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.

17. If you have installed GIT, create the 'gitadmin' user (which is already added in the site-options token `USERS_WITH_NO_EXPIRY_PASSWORD`) with site administrator rights in the TeamForge user interface.

18. If you have installed Git, integrate Gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

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
```

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

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

20. Run the following initial load jobs (ETL).

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 325.

### Do this on my.codesight.host

21. Install the Black Duck Code Sight license on the server where Black Duck Code Sight is installed. For more information, see [Install the Black Duck Code Sight license](#) on page 302.

22. 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.

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

23. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -v restart jboss
```

#### Do this on my.app.host

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

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

25. Restart the collabnet services.

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

#### Do this on my.codesight.host

26. Restart the Black Duck Code Sight service.

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

#### Do this on my.app.host

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

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

- c) Create a sample project.  
See [Create a TeamForge project](#).
- d) Write a welcome message to your site's users.  
See [Create a site-wide broadcast](#).

For specific instructions on installing Review Board, see: [Set up Review Board](#) on page 305

### Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see [TeamForge Orchestrate installation](#).

#### Install TeamForge 7.1 with SCM and Git integration on a separate server


In this option, we install SCM (Subversion, CVS) and GIT integrations on a separate server and other services on the main application server.

In this option, the following services run on the application server (we call this my.app.host).

- TeamForge Application Server
- Black Duck Code Sight Server
- Database Server (Operational DB and Reports DB)
- ETL Server
- Search Server (Indexer).

The following service runs on the SCM server. (We call this my.scmandgit.host)

- SCM Integration Server (Subversion and CVS)
- GIT Integration Server

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

Log in to the server as root.

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

1. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See [Platform specification for TeamForge 7.1](#) on page 356 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 server](#) on page 7 for details.

3. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

4. 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
```

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

```
zypper install teamforge-codesearch
```

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

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

a) Identify the servers and services running on them.

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


```
DOMAIN_localhost=my.app.domain.com
```

```
HOST_<my.scmangit.domain.com>=subversion cvs Gerrit
```

b) Configure the following settings if you are installing Black Duck Code Sight.

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

c) Configure the database and datamart settings.

 **Note:** For more information about configuring variables, see [site-options.conf](#) on page 383

```
DATABASE_TYPE=postgresql
```

```
DATABASE_USERNAME=ctfuser
```

```
DATABASE_NAME=ctfdb
```

```
DATABASE_READ_ONLY_USER=ctfrouser
```

```
REPORTS_DATABASE_USERNAME=ctfrptuser
```

```
REPORTS_DATABASE_NAME=ctfrptdb
```

```
REPORTS_DATABASE_READ_ONLY_USER=ctfrptrouser
```

```
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

 **Note:** The database name and username values are arbitrary alphanumeric strings.

- d) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to **Sauto\$**, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- DATABASE\_PASSWORD
- DATABASE\_READ\_ONLY\_PASSWORD
- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD
- BDCS\_ADMIN\_PASSWORD
- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


- e) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`;

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- f) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

```

• SSL=on
• SSL_CERT_FILE=
• SSL_KEY_FILE=
• SSL_CA_CERT_FILE=
• SSL_CHAIN_FILE=

```

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- g) If the token `REQUIRE_PASSWORD_SECURITY` is enabled, then set a value for the token, `PASSWORD_CONTROL_EFFECTIVE_DATE`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- h) 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.
- i) 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__}
```

- j) If you have LDAP set up for external authentication, you must set the `REQUIRE_USER_PASSWORD_CHANGE` site options token to false.
- k) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and Postgresql automatically.
- l) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

- 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.

 **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 username add this
token:
BDCS_ADMIN_USERNAME=<sysadmin>
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 are 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 the 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
```

- To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
  - If you are installing TeamForge through disconnected media, set the token **HELP\_AVAILABILITY=local**.
  - Save the `site-options.conf` file.
6. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```


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

```
export http_proxy=
```

**Do this on the SCM Server - my.scmangit.host**

**8.** Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See [Platform specification for TeamForge 7.1](#) on page 356 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.


**9.** Check your basic networking setup.

See [Set up networking for your TeamForge server](#) on page 7 for details.

**10.** Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)**11.** Install the TeamForge SCM and Git packages.

```
zypper install teamforge-scm teamforge-git
```

**12.** Copy the `site-options.conf` file from the application server to the SCM server in the directory `/opt/collabnet/teamforge-installer/7.1.0.0/conf`**13.** Modify the host token settings on the `site-options.conf` file.

 **Note:** If you choose not to use the application server's `site-options.conf` file, then don't forget to copy the value of `AUTO_DATA` token from the application server.

```
HOST_localhost=subversion cvs Gerrit
```

```
DOMAIN_localhost=my.scmangit.domain.com
```

```
HOST_my.app.domain.com=app database datamart etl indexer codesearch
```

**14.** Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

**15.** Set up the initial site data (bootstrap).


```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./bootstrap-data.sh
```

**Do the following on the application server - my.app.host****16.** Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./bootstrap-data.sh
```

**17.** Start TeamForge.

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

 **Note:** Startup can take several minutes, depending on the speed of the host system configuration. 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.

18. 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 302.

19. Run the following initial load jobs (ETL).

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 325.

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

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


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.
- 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:

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

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL path of the SCM integration server, for example, "`http://myint.box.net/svn/repos`", where `myint.box` is the server with the SCM integration server.

In addition, if you add a new integration server at some point later, you must run this `svn_cache.sh` script, (after creating the new integration server), on the TeamForge application 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 server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

**Do this on the SCM server - `my.scmandgit.host`**

24. If you have installed GIT, create the 'gitadmin' user (which is already added in the site-options token `USERS_WITH_NO_EXPIRY_PASSWORD`) with site administrator rights in the TeamForge user interface.

25. If you have installed Git, integrate gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

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
```

### Do the following on the application server - my.app.host

26. Restart the collabnet services.

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

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

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

b) 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.

c) Create a sample project.

See [Create a TeamForge project](#).

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

See [Create a site-wide broadcast](#).

For specific instructions on installing Review Board, see: [Set up Review Board](#) on page 305

### Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see [TeamForge Orchestrate installation](#).

### Install TeamForge 7.1 with Database and SCM on separate servers

In this option, we install the Database (Operational Database) and Datamart (Reporting Database) on the same server; SCM (Subversion and CVS) and Git on the second server, and other services on the application server.

In this option, the following services run on the application server (we call this my.app.host).


- TeamForge Application Server
- Black Duck Code Sight Server
- ETL Server
- Search Server (Indexer).

The following service runs on the database server. (We call this my.db.host)

- Database Server (Operational DB and Reports DB)

The following services run on the SCM server. (We call this my.scmndgit.host )

- SCM Integration Server (Subversion and CVS)
- GIT Integration Server

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

Log in to the server as root.

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

1. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See [Platform specification for TeamForge 7.1](#) on page 356 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 server](#) on page 7 for details.

3. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

4. Install the following application packages.

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

```
zypper install teamforge-app teamforge-etl
```

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

```
zypper install teamforge-codesearch
```

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

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

a) Identify the servers and services running on them.

```
HOST_localhost=app etl indexer
```

```
DOMAIN_localhost=my.app.domain.com
```


```
HOST_<my.db.domain.com>=database datamart
```

```
HOST_<my.scmangit.domain.com>=subversion cvs gerrit
```

b) Configure the following settings if you are installing Black Duck Code Sight.

```
HOST_localhost=app etl indexer codesearch
```

c) Configure the database and datamart settings.

 **Note:** For more information about configuring variables, see [site-options.conf](#) on page 383

```
DATABASE_TYPE=postgresql
```

```
DATABASE_USERNAME=ctfuser
```

```
DATABASE_NAME=ctfdb
```


```
DATABASE_READ_ONLY_USER=ctfrouser
```

```
REPORTS_DATABASE_USERNAME=ctfrptuser
```

```
REPORTS_DATABASE_NAME=ctfrptdb
```

```
REPORTS_DATABASE_READ_ONLY_USER=ctfrptrouser
```

```
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

 **Note:** The database name and username values are arbitrary alphanumeric strings.

- d) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to **\$auto\$**, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- DATABASE\_PASSWORD
- DATABASE\_READ\_ONLY\_PASSWORD
- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD
- BDCS\_ADMIN\_PASSWORD
- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


- e) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`;

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- f) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

- `SSL=on`
- `SSL_CERT_FILE=`
- `SSL_KEY_FILE=`
- `SSL_CA_CERT_FILE=`
- `SSL_CHAIN_FILE=`

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- g) If the token `REQUIRE_PASSWORD_SECURITY` is enabled, then set a value for the token, `PASSWORD_CONTROL_EFFECTIVE_DATE`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- h) 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.

- i) 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__}`
- j) If you have LDAP set up for external authentication, you must set the `REQUIRE_USER_PASSWORD_CHANGE` site options token to false.
- k) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and Postgresql automatically.
- l) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

- 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.

 **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 username add this token:  
`BDCS_ADMIN_USERNAME=<sysadmin>`  
 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 are 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 the 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
```

- o) To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
- p) If you are installing TeamForge through disconnected media, set the token `HELP_AVAILABILITY=local`.
- q) Save the `site-options.conf` file.

#### 6. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

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

```
export http_proxy=
```

#### Do this on the database server - my.db.host

#### 8. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See [Platform specification for TeamForge 7.1](#) on page 356 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.

#### 9. Check your basic networking setup.

See [Set up networking for your TeamForge server](#) on page 7 for details.

#### 10. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

#### 11. Install the TeamForge database packages.

```
zypper install teamforge-database
```

#### 12. Copy the `site-options.conf` file from the application server to the database server in the directory `/opt/collabnet/teamforge-installer/7.1.0.0/conf`

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

- 👉 **Note:** If you choose not to use the application server's `site-options.conf` file, then don't forget to copy the value of `AUTO_DATA` token from the application server.

```
HOST_localhost=database datamart
```

```
DOMAIN_localhost=my.db.domain.com
```

```
HOST_my.app.domain.com=app etl indexer codesearch
```

```
HOST_<my.scmangit.domain.com>=subversion cvs gerrit
```

#### 14. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

#### Do this on the SCM Server - my.scmangit.host

#### 15. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See [Platform specification for TeamForge 7.1](#) on page 356 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.

#### 16. Check your basic networking setup.

See [Set up networking for your TeamForge server](#) on page 7 for details.

#### 17. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

#### 18. Install the TeamForge SCM and Git packages.

```
zypper install teamforge-scm teamforge-git
```

#### 19. Copy the `site-options.conf` file from the application server to the SCM server in the directory `/opt/collabnet/teamforge-installer/7.1.0.0/conf`

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

- 👉 **Note:** If you choose not to use the application server's `site-options.conf` file, then don't forget to copy the value of `AUTO_DATA` token from the application server.

```
HOST_localhost=subversion cvs gerrit
```

```
DOMAIN_localhost=my.scm.domain.com
```

```
HOST_my.app.domain.com=app etl indexer codesearch
```

```
HOST_<my.db.domain.com>=database datamart
```

#### 21. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

#### 22. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./bootstrap-data.sh
```


#### Do the following on the application server - my.app.host

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

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./bootstrap-data.sh
```

#### 24. Start TeamForge.

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

 **Note:** Startup can take several minutes, depending on the speed of the host system configuration. 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. 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 302.

26. Run the following initial load jobs (ETL).

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 325.

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

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


28. 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.
- 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
```

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

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

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```



Provide a repository base URL path of the SCM integration server, for example, "http://myint.box.net/svn/repos", where `myint.box` is the server with the SCM integration server.

In addition, if you add a new integration server at some point later, you must run this `svn_cache.sh` script, (after creating the new integration server), on the TeamForge application server.

30. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

#### Do this on the SCM server - `my.scmandgit.host`

31. If you have installed GIT, create the 'gitadmin' user (which is already added in the site-options token `USERS_WITH_NO_EXPIRY_PASSWORD`) with site administrator rights in the TeamForge user interface.

32. If you have installed Git, integrate gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

- 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
```

#### Do the following on the application server - `my.app.host`

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

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

34. 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
```

35. 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
```

#### Do the following on the application server - `my.app.host`

36. Restart the collabnet services.

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

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

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

- c) Create a sample project.

See [Create a TeamForge project](#).

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

See [Create a site-wide broadcast](#).

For specific instructions on installing Review Board, see: [Set up Review Board](#) on page 305

## Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see [TeamForge Orchestrate installation](#).

### Install Git integration on a separate server

In this option, we install the GIT integration services on a separate server.

Log in to the server as root.

#### Do this on the Git server. We'll call this `my.git.host`.

1. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See [Platform specification for TeamForge 7.1](#) on page 356 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 server](#) on page 7 for details.

3. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

4. Install the Git packages.

```
zypper install teamforge-git
```

5. Configure the token settings for Git in the `site-options.conf` file.

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

a) Modify the host token settings.



**Note:** If you choose not to use the application server's `site-options.conf` file, then don't forget to copy the value of `AUTO_DATA` token from the application server.

```
HOST_localhost=gerrit
```

```
DOMAIN_localhost=my.git.domain.com
```

```
HOST_my.app.domain.com=app database datamart etl indexer subversion cvs
```

b) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and Postgresql automatically.

c) To enable the history protection feature of TeamForge Git integration, set the `GERRIT_FORCE_HISTORY_PROTECTION=true`. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396

- d) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

- `SSL=on`
- `SSL_CERT_FILE=`
- `SSL_KEY_FILE=`
- `SSL_CA_CERT_FILE=`
- `SSL_CHAIN_FILE=`



**Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- e) Save the `site-options.conf` file.

#### 6. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

7. Create the 'gitadmin' user (which is already added in the `site-options` token `USERS_WITH_NO_EXPIRY_PASSWORD`) with site administrator rights in the TeamForge user interface.
8. If you have installed Git, integrate gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

- 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](#).

### 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 server or on different servers, in whatever combination works best for your conditions.

In principle, a multi-server 7.1 site can have its services running in a wide variety of combinations on an undefined number of servers. However, real-world sites tend to follow one of the following patterns, depending on the specific needs of the community of site users.

### Install TeamForge 7.1 with Oracle database on a separate server


In this option, we install the Oracle database (Operational database and Reports database) on a separate server and other services on the main application server.

In this option, the following services run on the application server (we call this my.app.host).

- TeamForge Application Server
- Black Duck Code Sight Server
- ETL Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer).

The following service runs on the database server. (We call this my.db.host)

- Database Server (Operational DB and Reports DB)

 **Note:** If either of the remote servers (the data server or the source code server) is not under your direct control, check with the Database Administrator to make sure that you can carry out these instructions on that server.

1. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See [Platform specification for TeamForge 7.1](#) on page 356 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 server](#) on page 7 for details.

3. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

4. Install the following application packages.

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


```
zypper install teamforge-app teamforge-etl teamforge-scm
```

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

```
zypper install teamforge-codesearch
```

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

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
cp conf/site-options-advanced.conf conf/site-options.conf
```

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

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

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

- a) Configure the HOST token.

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


```
DOMAIN_localhost=my.app.domain.com
```

```
HOST_<my.db.host>=database datamart
```

- b) Configure the following settings if you are installing Black Duck Code Sight.

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

- c) Configure the database and datamart settings.

 **Note:** For more information about configuring variables, see [site-options.conf](#) on page 383

```
DATABASE_TYPE=oracle
```

```
DATABASE_USERNAME=sitedatabaseusername
```

```
DATABASE_PASSWORD=sitedatabasepwd
```

```
DATABASE_READ_ONLY_USER=sitedatabasereadonlyusername
```

```
DATABASE_READ_ONLY_PASSWORD=sitedatabasereadonlyuserpwd
```

```
DATABASE_NAME=sitedatabaseinstancename
```

```
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
```

```
DATABASE_SERVICE_NAME=
```

```
REPORTS_DATABASE_SERVICE_NAME=
```

 **Tip:** To find the value for the token `DATABASE_SERVICE_NAME` log in to your Oracle server and execute this command.

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

Find the value of the `SERVICE_NAME` in the output and use this value for the `DATABASE_SERVICE_NAME` in the `site-options.conf` file.


#### d) Password Obfuscation

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`).

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- e) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

```

• SSL=on
• SSL_CERT_FILE=
• SSL_KEY_FILE=
• SSL_CA_CERT_FILE=
• SSL_CHAIN_FILE=

```

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- f) If the token `REQUIRE_PASSWORD_SECURITY` is enabled, then set a value for the token, `PASSWORD_CONTROL_EFFECTIVE_DATE`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- g) 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.
- 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) If you have LDAP set up for external authentication, you must set the `REQUIRE_USER_PASSWORD_CHANGE` site options token to false.
- j) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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) 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.

-  **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 username add this token:

```
BDCS_ADMIN_USERNAME=<sysadmin>
```

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 are 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 the 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
```

m) If you are installing TeamForge through disconnected media, set the token `HELP_AVAILABILITY=local`.

n) Save the `site-options.conf` file.


## 7. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
```

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

8. If you are installing on a server that is behind a proxy, unset the `http_proxy` token.

```
export http_proxy=
```

9.  **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 localinstall <path to oracle client rpm>
```

10. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

11. Copy the Oracle datamart setup script from `/opt/collabnet/teamforge/runtime/scripts` to the `/tmp` directory of `my.db.host`.

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

#### Do this on the database server `my.db.host`

12. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See [Platform specification for TeamForge 7.1](#) on page 356 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. Check your basic networking setup.

See [Set up networking for your TeamForge server](#) on page 7 for details.


14. Copy the Oracle datamart setup script.

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

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

See [Set up an Oracle database](#) on page 263 for help.

16. Create the reporting user and schema.

 **Note:** Skip this step if you have already set up the datamart setup in the Oracle database. Your responses to the script's prompts must match the values of the equivalent variables in the `site-options.conf` file on `my.app.server`.


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

#### Do this on the TeamForge Application Server (`my.app.host`)

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

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

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
```

19. 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
```

20. 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
```

21. 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.

22. 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 302.

23. Run the following initial load jobs (ETL).

- 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 325.

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

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

25. 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.

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

26. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

27. Restart the collabnet services.

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

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

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

- b) 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.

- c) Create a sample project.

See [Create a TeamForge project](#).

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

See [Create a site-wide broadcast](#).

For specific instructions on installing Review Board, see: [Set up Review Board](#) on page 305

## Installing TeamForge Orchestrate


To install TeamForge Orchestrate, see [TeamForge Orchestrate installation](#).

## Verify your TeamForge 7.1 installation

Congratulations: you have just installed your TeamForge 7.1 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 272 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 260.


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 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 269.

 **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 7.1 site](#) on page 256 to help keep your site going.

## Uninstall TeamForge 7.1

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 7.1, see [Hardware and software requirements for CollabNet TeamForge 7.1 on a virtual machine](#) on page 360.

### Get TeamForge 7.1 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](http://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 7.1 installer for VMware Player

3. Contact the [CollabNet Support](#) and download the TeamForge installer zip file.

4. Unzip the `TeamForge-7.1.0.0-VMware.zip` file.

#### Installing the TeamForge 7.1 on VMware Player

5. Click **Open a Virtual Machine**.

6. Browse and select the `TeamForge-71.ova` file from where you unzipped the `TeamForge-7.1.0.0-VMware.zip` file earlier.

7. Click **Import**.

Once successfully imported, you can see a new virtual machine named "**TeamForge-71**" 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-71** virtual machine and click **Play Virtual Machine**.

It may take a few minutes to start the **TeamForge-71** 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](#).

## 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 7.1 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 7.1 on a VMware ESXi server:

1. Contact the [CollabNet Support](#) and download the TeamForge installer zip file.
2. Unzip the `TeamForge-7.1.0.0-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-71.ova` file from where you unzipped the `TeamForge-7.1.0.0-VMware-xxx.zip` file earlier.
6. Click **Next**.  
The **OVF Template Details** wizard appears.
7. Click **Next**.
8. Type a name, for example "TeamForge-7.1" 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-7.1" virtual machine.
13. Select the "TeamForge-7.1" 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](#).

## 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 260.

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 7.1

To remove TeamForge completely, use the YUM utility.

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

1. Stop TeamForge.

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

2. Run yum to remove TeamForge.

```
yum erase TeamForge-installer
```

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

## Upgrade to TeamForge 7.1

---

When you upgrade to TeamForge 7.1, you can have Black Duck Code Sight on the same server as the main TeamForge application, or on a separate one. Choose the instructions that fit your case.

### Plan your upgrade to TeamForge 7.1

As the first step in upgrading to TeamForge 7.1, consider some key questions that will affect how your new site works.

1. Where is everything?

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

2. Provide site-wide reporting?

TeamForge 6.1.x and later let site administrators track user logins.

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

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

3. Complete site re-indexing required?

- If you are upgrading from TeamForge 6.1.1 or later versions to TeamForge 7.1, you can run the `indexupgrade.py` script, which converts the Lucene 3.x indices to Lucene 4.4 format. For more information about the `indexupgrade.py` script, see [Upgrade TeamForge 7.1 search index to Lucene 4.x format](#) on page 318. However, if you choose to re-index data, it takes a lot of time and the search service would have to be down till then. Instead, if you choose to upgrade your existing indices, you can convert your site's search indices to Lucene 4.x format quickly using the `indexupgrade.py` script with less downtime of the search service.
- If you are upgrading from TeamForge 6.1 or earlier versions to TeamForge 7.1, you must re-index your site completely. This could be time consuming and depends on the size of data.

4. "Dedicated" and "Advanced" installation types

The type of TeamForge installation you have makes a difference for how you upgrade and patch the site. If your site is a dedicated site, you'll be able to skip some of the steps outlined here. If you don't know whether your site was originally installed as dedicated or advanced, here's how you can find out: [Is my TeamForge site "dedicated" or "advanced"?](#) on page 253

5. Branding changes?

Every release of TeamForge can bring changes to the look and feel of the product. TeamForge 7.1 is no exception. If you have edited files in your site's branding repository (that's how you customize the look and feel of the product), you must download the new branding package and check into your branding repository the new versions of any files you have edited. See [Customize anything on your site](#) for instructions.

6. 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 321 for recommendations from CollabNet's performance team on optimizing PostgreSQL for different conditions.

## 7. Upgraded JDK

TeamForge 7.1 uses JDK 1.7.0\_40. If you are upgrading on the same server, and that server has an older JDK version, the TeamForge 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.


## 8. Lucene 4.4 upgrade

TeamForge 7.1 uses Lucene 4.4. Post upgrade to TeamForge 7.1, you must run the `indexupgrade.py` script to convert your site's existing indices to Lucene 4.x format. Running this script saves you time, which otherwise must be spent in reindexing your site's indices. Converting indices to Lucene 4.x format also improves your site's search performance. For more information, see:

- [Upgrade TeamForge 7.1 search index to Lucene 4.x format](#)
- [indexupgrade.py](#) on page 378

## Upgrade to TeamForge 7.1 on Red Hat/CentOS

You can upgrade to TeamForge 7.1 from TeamForge 7.0. 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 installs the CollabNet TeamForge site in the default directory `"/opt/collabnet/teamforge"`.

 **Note:**

- If you are upgrading from TeamForge 6.1.1 or later versions to TeamForge 7.1, you can run the `indexupgrade.py` script, which converts the Lucene 3.x indices to Lucene 4.4 format. For more information about the `indexupgrade.py` script, see [Upgrade TeamForge 7.1 search index to Lucene 4.x format](#) on page 318. However, if you choose to re-index data, it takes a lot of time and the search service would have to be down till then. Instead, if you choose to upgrade your existing indices, you can convert your site's search indices to Lucene 4.x format quickly using the `indexupgrade.py` script with less downtime of the search service.
- If you are upgrading from TeamForge 6.1 or earlier versions to TeamForge 7.1, you must re-index your site completely. This could be time consuming and depends on the size of data.

### Upgrade to TeamForge 7.1 - All Services on the same server

In this procedure, we'll assume that you are upgrading on the same server where your existing TeamForge 7.0 site is running. It is possible to simultaneously upgrade and move your site to a new hardware. However, since we are working with a dedicated installation, the priority here is to keep things as simple and quick as possible.

In this option, the following services run on the application server (we call this `my.app.host`).

- TeamForge Application Server
- Black Duck Code Sight Server
- Database Server (Operational DB and Reports DB)
- ETL Server
- GIT Integration Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer).

**Log in to the server as root.**


**Do the following on the application server - `my.app.host`**

#### 1. Stop TeamForge.


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



## 2. Migrate your PostgreSQL to the latest version supported by TeamForge 7.1.

 **Important:** You must have RHEL/CentOS 6.5 or later to run PGTurant. Upgrade the operating system packages (`yum upgrade`) before running PGTurant on sites running on RHEL/CentOS versions 6.4 or earlier.

a) Install PGTurant. See [Upgrade PostgreSQL using PGTurant](#) on page 256 for more information.

 **Important:** If your TeamForge site has no internet access, contact the [CollabNet Support](#), get the `pgturant-8.0.0.0-x.noarch.rpm` package and unpack the RPM.

```
rpm -ivh pgturant-8.0.0.0-x.noarch.rpm
yum install pgturant -y
```

b) Upgrade TeamForge PostgreSQL data directory to PostgreSQL 9.2.


```
cd /opt/collabnet/pgturant/bin/
./pgturant -s /var/lib/pgsql/9.0/data -d /var/lib/pgsql/9.2 -u 9.2 -m
```

## 3. Start the PostgreSQL service, run the `analyze_new_cluster.sh` script and stop PostgreSQL.

```
/etc/init.d/postgresql-9.2 start
su - postgres -c "/var/lib/pgsql/9.2/analyze_new_cluster.sh"
/etc/init.d/postgresql-9.2 stop
```

4. [Click here](#) only if your reporting database is running on a separate port.

5. Back up the file system data.

 **Tip:** `/tmp/backup_dir` is just an example. You can use any directory or partition you prefer to store your backup files.

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

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

```
cp -Rpfv /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 -Rpfv /gitroot /tmp/backup_dir
cp -Rpfv /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
backup_dir/gerrit
```

b) Back up your SSH keys, if any.

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

## 6. Move the collabnet repository of the older version of TeamForge.

```
mv /etc/yum.repos.d/collabnet-7.0.0.x.repo /etc/yum.repos.d/
collabnet-7.0.0.x.repo.cn_backup
```

 **Note:** Replace "x" with the appropriate patch release number if applicable.

## 7. Run the following commands to upgrade Red Hat/CentOS to the latest version.

- a) Remove the neon-devel package if you are upgrading from Teamforge 6.2.

```
yum erase neon-devel -y
yum erase subversion-devel -y
```

- b) Upgrade the operating system packages.

```
yum upgrade
```

8. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

- a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

- b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.


- c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```

See [Set up SELINUX](#) on page 271 to have TeamForge to run in SELinux mode after completing the installation or upgrade.


9. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

10. Uninstall the TeamForge CLI add-on (if it is already installed).

-  **Note:** Skip this step if you are upgrading from TeamForge 6.2

```
cd /opt/collabnet/teamforge/add-ons/teamforge_cli_server
./install --uninstall
yum erase teamforge_cli_server
```

11. Uninstall the PostgreSQL RPMs.

-  **Note:** When prompted, you must select the option to remove all the dependent packages.

```
yum erase postgresql90-libs postgresql90-docs postgresql90-server
postgresql90
```

12. 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 the Black Duck Code Sight packages run the following command:

```
yum install teamforge-codesearch
```

13. In the `site-options.conf` file, make sure you do the following.

-  **Note:** Back up your `site-options.conf` file before making any changes.

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

- a) Update the host name and domain name, if required.

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

```
DOMAIN_localhost=my.app.domain.com
```

Configure the following settings if you are installing Git.

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

Configure the following settings if you are installing Black Duck Code Sight.

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

- b) Configure the `JAVA_HOME` token for TeamForge.

```
JAVA_HOME=/usr/java/jdk1.7.0_40
```

- c) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

- `SSL=on`
- `SSL_CERT_FILE=`
- `SSL_KEY_FILE=`
- `SSL_CA_CERT_FILE=`
- `SSL_CHAIN_FILE=`

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- 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`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- f) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to `$auto$`, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- `DATABASE_PASSWORD`
- `DATABASE_READ_ONLY_PASSWORD`
- `REPORTS_DATABASE_PASSWORD`
- `REPORTS_DATABASE_READ_ONLY_PASSWORD`
- `ETL_SOAP_SHARED_SECRET`
- `JAMES_ADMIN_PASSWORD`
- `BDCS_ADMIN_PASSWORD`
- `MIRROR_DATABASE_PASSWORD` (applicable only if you are mirroring your database)


- g) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF};`.

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- h) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and Postgresql automatically.
- i) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

- 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`

 **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 username add this token:  
`BDCS_ADMIN_USERNAME=<sysadmin>`  
 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) 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 396
- m) If you have LDAP set up for external authentication, you must set the “**REQUIRE\_USER\_PASSWORD\_CHANGE**” site options token to false.
- n) If you are installing TeamForge through disconnected media, set the token **HELP\_AVAILABILITY=local**.
- o) Ensure to set the token **SELINUX\_SETUP=false** temporarily in the `site-options.conf` file.
- p) Save the `site-options.conf` file.

14. Run the following command to remove the pagespeed cache.

```
cd /opt/collabnet/teamforge/cache
rm -rf pagespeed
```

15. Back up the old TeamForge runtime directory.


```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

16. Recreate the runtime environment.


```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

17. 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.

18. Convert your site data to work with TeamForge 7.1.


 **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 7.1 site.

19. Run the following script to upgrade the [index to Lucene 4.x format](#).

 **Note:** You must back up the existing search index directory before running this script. Refer to the `SEARCH_INDEX_LOCATION` token in the `runtime-options.conf` file to know the search index directory location.

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

20. Run the following script to upgrade the Subversion working copies.

```
/opt/collabnet/teamforge/runtime/scripts/svn-upgrade-working-copies.sh
```

21. 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.

22. If you are upgrading from TeamForge 7.0, run the `post-install.py` script.


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

23. If you are upgrading from TeamForge 6.2 or earlier versions and the CLI reports are already configured, follow these steps:

a) Back up the CLI reports.

```
mkdir /root/backup
cd /root/backup
backup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> find . -name .svn | xargs rm -rf
```

b) Remove the cli/reports folder from the branding repository.

 **Note:** Do a SVN delete to remove this folder from the repository.

```
mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> svn delete reports
cli> svn commit -m "To delete the old CLI reports folder"
```

c) Run the post-install.py script.

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

d) Commit the backup 'pkg' folder which is available in this location branding/cli/custom-reports/

```
mkdir /root/restore
cd /root/restore
restore> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
mkdir custom-reports
cp -avx /root/backup/pkg custom-reports
svn add custom-reports
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>
```

When prompted, enter the user name (<admin\_username>) and password (<admin\_password>)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```

24. If you have installed Git, integrate gerrit by running the post-install.py script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.

- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

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. Install the Black Duck Code Sight license on the server where Black Duck Code Sight is installed. For more information see [these instructions](#).

26. 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.

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

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

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

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL path of the SCM integration server, for example, "`http://myint.box.net/svn/repos`", where `myint.box` is the server with the SCM integration server.

In addition, if you add a new integration server at some point later, you must run this `svn_cache.sh` script, (after creating the new integration server), on the TeamForge application server.

28. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -v restart jboss
```

29. Run the following initial load jobs (ETL).

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 325.

30. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.1.


a) Log into your site as the administrator.

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


See [Customize anything on your site](#).

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

See [Create a site-wide broadcast](#).

 **Important:** Do not delete the `teamforge-installer/7.1.0.0` directory. You will need it for future maintenance and upgrades.

31. Remove the repository and the file system backup files from `/tmp/backup_dir` directory after the TeamForge site is up and running as expected.

 **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](#).

To upgrade Review Board [see these instructions](#).

To run TeamForge in SELINUX enabled mode, see [Set up SELINUX](#) on page 271


### Upgrade to TeamForge 7.1 - Database and Datamart on a separate server

In this procedure, we'll assume that you are upgrading on the same server where your existing TeamForge 7.0 site is running. It is possible to simultaneously upgrade and move your site to a new hardware. However, since we are working with a dedicated installation, the priority here is to keep things as simple and quick as possible.

In this option, the following services run on the TeamForge Application Server (We call this `my.app.host`).

- TeamForge Application Server
- Black Duck Code Sight Server
- ETL Server
- GIT Integration Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer)

Both the operational and reports databases run on the Database Server (Operational DB and Reports DB). (We call this `my.db.host`)

 **Note:** For the ETL service to run as expected in a distributed TeamForge installation, all servers must have the same time zone.


### Log in to the server as root.

#### Do the following on the application server - `my.app.host`

1. Stop TeamForge.

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

2. Back up the file system data.

 **Tip:** `/tmp/backup_dir` is just an example. You can use any directory or partition you prefer to store your backup files.

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

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

```
mkdir -p /tmp/backup_dir
```




```
cp -Rpfv /svnroot /sf-svnroot /cvsroot /opt/collabnet/teamforge/var /
tmp/backup_dir
```

- b) If GIT integration is enabled, do the following:


```
mkdir /tmp/backup_dir/gerrit
cp -Rpfv /gitroot /tmp/backup_dir
cp -Rpfv /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
backup_dir/gerrit
```

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

3. Migrate your PostgreSQL to the latest version supported by TeamForge 7.1.

 **Important:** You must have RHEL/CentOS 6.5 or later to run PGTurant. Upgrade the operating system packages (`yum upgrade`) before running PGTurant on sites running on RHEL/CentOS versions 6.4 or earlier.

- a) Install PGTurant. See [Upgrade PostgreSQL using PGTurant](#) on page 256 for more information.

 **Important:** If your TeamForge site has no internet access, contact the [CollabNet Support](#), get the `pgturant-8.0.0.0-x.noarch.rpm` package and unpack the RPM.

```
rpm -ivh pgturant-8.0.0.0-x.noarch.rpm
yum install pgturant -y
```

- b) Upgrade TeamForge PostgreSQL data directory to PostgreSQL 9.2.

```
cd /opt/collabnet/pgturant/bin/
./pgturant -s /var/lib/pgsql/9.0/data -d /var/lib/pgsql/9.2 -u 9.2 -m
```


4. Start the PostgreSQL service, run the `analyze_new_cluster.sh` script and stop PostgreSQL.

```
/etc/init.d/postgresql-9.2 start
su - postgres -c "/var/lib/pgsql/9.2/analyze_new_cluster.sh"
/etc/init.d/postgresql-9.2 stop
```


5. [Click here](#) only if your reporting database is running on a separate port.

#### Do this on the database server - my.db.host

6. Migrate your PostgreSQL to the latest version supported by TeamForge 7.1.

 **Important:** You must have RHEL/CentOS 6.5 or later to run PGTurant. Upgrade the operating system packages (`yum upgrade`) before running PGTurant on sites running on RHEL/CentOS versions 6.4 or earlier.

- a) Install PGTurant. See [Upgrade PostgreSQL using PGTurant](#) on page 256 for more information.

 **Important:** If your TeamForge site has no internet access, contact the [CollabNet Support](#), get the `pgturant-8.0.0.0-x.noarch.rpm` package and unpack the RPM.

```
rpm -ivh pgturant-8.0.0.0-x.noarch.rpm
yum install pgturant -y
```

- b) Upgrade TeamForge PostgreSQL data directory to PostgreSQL 9.2.

```
cd /opt/collabnet/pgturant/bin/
./pgturant -s /var/lib/pgsql/9.0/data -d /var/lib/pgsql/9.2 -u 9.2 -m
```

7. Start the PostgreSQL service, run the `analyze_new_cluster.sh` script and stop PostgreSQL.

```
/etc/init.d/postgresql-9.2 start
su - postgres -c "/var/lib/pgsql/9.2/analyze_new_cluster.sh"
/etc/init.d/postgresql-9.2 stop
```

8. [Click here](#) only if your reporting database is running on a separate port.

#### Do this on the application server - my.app.host

9. Move the collabnet repository of the older version of TeamForge.

```
mv /etc/yum.repos.d/collabnet-7.0.0.x.repo /etc/yum.repos.d/
collabnet-7.0.0.x.repo.cn_backup
```

 **Note:** Replace "x" with the appropriate patch release number if applicable.

10. Run the following commands to upgrade Red Hat/CentOS to the latest version.

- a) Remove the neon-devel package if you are upgrading from Teamforge 6.2.

```
yum erase neon-devel -y
yum erase subversion-devel -y
```

- b) Upgrade the operating system packages.

```
yum upgrade
```

11. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

- a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

- b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.


- c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```

See [Set up SELINUX](#) on page 271 to have TeamForge to run in SELinux mode after completing the installation or upgrade.


12. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

13. Uninstall the TeamForge CLI add-on (if it is already installed).

 **Note:** Skip this step if you are upgrading from TeamForge 6.2

```
cd /opt/collabnet/teamforge/add-ons/teamforge_cli_server
./install --uninstall
yum erase teamforge_cli_server
```

14. If GIT is enabled, uninstall the PostgreSQL database used by Gerrit as PostgreSQL may be upgraded later during the installation.

 **Note:** When prompted, you must select the option to remove all the dependent packages.

```
yum erase postgresql90-libs postgresql90-docs postgresql90-server
postgresql90
```

15. Install the following application packages.

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

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


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

```
yum install teamforge-git
```

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

```
yum install teamforge-codesearch
```

16. Update your `site-options.conf` file.

 **Important:** Back up your `site-options.conf` file before making any changes.

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

- a) Update the host name and domain name, if required.

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

```
DOMAIN_localhost=my.app.domain.com
```

```
HOST_my.db.host=database datamart
```

Add "gerrit" to Host\_localhost if you are installing Git.

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

Add "codesearch" to Host\_localhost if you are installing Black Duck Code Sight.

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

- b) Configure the `JAVA_HOME` token for TeamForge.

```
JAVA_HOME=/usr/java/jdk1.7.0_40
```

- c) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

- `SSL=on`
- `SSL_CERT_FILE=`
- `SSL_KEY_FILE=`
- `SSL_CA_CERT_FILE=`
- `SSL_CHAIN_FILE=`

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- 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`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- f) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to `$auto$`, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- DATABASE\_PASSWORD
- DATABASE\_READ\_ONLY\_PASSWORD
- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD
- BDCS\_ADMIN\_PASSWORD
- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


#### g) Password Obfuscation

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF};`.

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- h) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and PostgreSQL automatically.
- i) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

- 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
```

 **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 username add this
token:
BDCS_ADMIN_USERNAME=<sysadmin>
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) To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
- m) If you have LDAP set up for external authentication, you must set the **"REQUIRE\_USER\_PASSWORD\_CHANGE"** site options token to false.
- n) If you are installing TeamForge through disconnected media, set the token **HELP\_AVAILABILITY=local**.
- o) Ensure to set the token **SELINUX\_SETUP=false** temporarily in the `site-options.conf` file.
- p) Save the `site-options.conf` file.

17. Run the following command to remove the pagespeed cache.

```
cd /opt/collabnet/teamforge/cache
rm -rf pagespeed
```

18. Back up the old TeamForge runtime directory.


```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

19. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

20. 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 database server - my.db.host**

21. Move the collabnet repository of the older version of TeamForge.

```
mv /etc/yum.repos.d/collabnet-7.0.0.x.repo /etc/yum.repos.d/
collabnet-7.0.0.x.repo.cn_backup
```

 **Note:** Replace "x" with the appropriate patch release number if applicable.

22. Run the following commands to upgrade Red Hat/CentOS to the latest version.

- a) Remove the neon-devel package if you are upgrading from Teamforge 6.2.

```
yum erase neon-devel -y
yum erase subversion-devel -y
```

- b) Upgrade the operating system packages.

```
yum upgrade
```

23. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

- a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

- b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.
- c) If not disabled, run the following command to disable SELinux.


```
setenforce 0
```

24. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

25. If the PostgreSQL database is running locally, stop the PostgreSQL service.

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

26. Uninstall the PostgreSQL RPMs.


 **Note:** When prompted, you must select the option to remove all the dependent packages.

```
yum erase postgresql90-libs postgresql90-docs postgresql90-server
postgresql90
```

27. Install the TeamForge database packages.

```
yum install teamforge-database
```

28. Copy the `site-options.conf` file from **my.app.host** and modify the token settings.

 **Note:** If you choose to use the old `site-options.conf` file, don't forget to copy the `AUTO_DATA` token from the application server.

```
HOST_localhost=database datamart
```

```
DOMAIN_localhost=my.db.domain.com
```

```
HOST_my.app.host=app etl indexer subversion cvs
```

29. Back up the old TeamForge runtime directory.


```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

30. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

**Do this on the application server - my.app.host**

31. Convert your site data to work with TeamForge 7.1.


 **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 7.1 site.

32. Run the following script to upgrade the [index to Lucene 4.x format](#).

 **Note:** You must back up the existing search index directory before running this script. Refer to the `SEARCH_INDEX_LOCATION` token in the `runtime-options.conf` file to know the search index directory location.

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

33. Run the following script to upgrade the Subversion working copies.

```
/opt/collabnet/teamforge/runtime/scripts/svn-upgrade-working-copies.sh
```

34. 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.

35. If you are upgrading from TeamForge 7.0, run the `post-install.py` script.


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

36. If you are upgrading from TeamForge 6.2 or earlier versions and the CLI reports are already configured, follow these steps:

a) Back up the CLI reports.

```
mkdir /root/backup
cd /root/backup
backup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> find . -name .svn | xargs rm -rf
```

b) Remove the `cli/reports` folder from the branding repository.

 **Note:** Do a SVN delete to remove this folder from the repository.

```
mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> svn delete reports
cli> svn commit -m "To delete the old CLI reports folder"
```

c) Run the `post-install.py` script.

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

d) Commit the backup 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```
mkdir /root/restore
cd /root/restore
restore> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
mkdir custom-reports
cp -avx /root/backup/pkg custom-reports
svn add custom-reports
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>
```

When prompted, enter the user name (<admin\_username>) and password (<admin\_password>)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```

37. If you have installed Git, integrate Gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

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
```

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.

```
/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:

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

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL path of the SCM integration server, for example, "`http://myint.box.net/svn/repos`", where `myint.box` is the server with the SCM integration server.

In addition, if you add a new integration server at some point later, you must run this `svn_cache.sh` script, (after creating the new integration server), on the TeamForge application 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 server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

41. Run the following initial load jobs (ETL).

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 325.


42. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.1.



- a) Log into your site as the administrator.
- b) If your site has custom branding, verify that your branding changes still work as intended.  
See [Customize anything on your site](#).
- c) Let your site's users know they've been upgraded.  
See [Create a site-wide broadcast](#).

 **Important:** Do not delete the `teamforge-installer/7.1.0.0` directory. You will need it for future maintenance and upgrades.

43. Remove the repository and the file system backup files from `/tmp/backup_dir` directory after the TeamForge site is up and running as expected.

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

To install TeamForge Orchestrate, see [TeamForge Orchestrate installation](#).

To upgrade Review Board [see these instructions](#).

To run TeamForge in SELINUX enabled mode, see [Set up SELINUX](#) on page 271

### Upgrade to TeamForge 7.1 - Reporting services on a separate server


In this procedure, we'll assume that you are upgrading on the same server where your existing TeamForge 7.0 site is running. It is possible to simultaneously upgrade and move your site to a new hardware. However, since we are working with a dedicated installation, the priority here is to keep things as simple and quick as possible.

In this option, the following services run on the application server (we call this `my.app.host`).

- TeamForge Application Server
- Black Duck Code Sight Server
- Database Server (Operational DB)
- GIT Integration Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer)

The following service runs on the database server. (We call this `my.reports.host`)

- Database server (reports db)
- ETL server

 **Note:** For the ETL service to run as expected in a distributed TeamForge installation, all servers must have the same time zone.


#### Log in to the server as root.

#### Do the following on the application server - `my.app.host`


1. Stop TeamForge.

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

2. Migrate your PostgreSQL to the latest version supported by TeamForge 7.1.

 **Important:** You must have RHEL/CentOS 6.5 or later to run PGTurant. Upgrade the operating system packages (`yum upgrade`) before running PGTurant on sites running on RHEL/CentOS versions 6.4 or earlier.

- a) Install PGTurant. See [Upgrade PostgreSQL using PGTurant](#) on page 256 for more information.

 **Important:** If your TeamForge site has no internet access, contact the [CollabNet Support](#), get the `pgturant-8.0.0.0-x.noarch.rpm` package and unpack the RPM.

```
rpm -ivh pgturant-8.0.0.0-x.noarch.rpm
```

```
yum install pgturant -y
```


- b) Upgrade TeamForge PostgreSQL data directory to PostgreSQL 9.2.

```
cd /opt/collabnet/pgturant/bin/
./pgturant -s /var/lib/pgsql/9.0/data -d /var/lib/pgsql/9.2 -u 9.2 -m
```

3. Start the PostgreSQL service, run the `analyze_new_cluster.sh` script and stop PostgreSQL.

```
/etc/init.d/postgresql-9.2 start
su - postgres -c "/var/lib/pgsql/9.2/analyze_new_cluster.sh"
/etc/init.d/postgresql-9.2 stop
```

4. Back up the file system data.

 **Tip:** `/tmp/backup_dir` is just an example. You can use any directory or partition you prefer to store your backup files.

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

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

```
cp -Rpfv /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 -Rpfv /gitroot /tmp/backup_dir
cp -Rpfv /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
backup_dir/gerrit
```


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

#### Do this on the reporting server - `my.reports.host`

5. Migrate your PostgreSQL to the latest version supported by TeamForge 7.1.

 **Important:** You must have RHEL/CentOS 6.5 or later to run PGTurant. Upgrade the operating system packages (`yum upgrade`) before running PGTurant on sites running on RHEL/CentOS versions 6.4 or earlier.

- a) Install PGTurant. See [Upgrade PostgreSQL using PGTurant](#) on page 256 for more information.

 **Important:** If your TeamForge site has no internet access, contact the [CollabNet Support](#), get the `pgturant-8.0.0.0-x.noarch.rpm` package and unpack the RPM.

```
rpm -ivh pgturant-8.0.0.0-x.noarch.rpm
yum install pgturant -y
```

- b) Upgrade TeamForge PostgreSQL data directory to PostgreSQL 9.2.

```
cd /opt/collabnet/pgturant/bin/
./pgturant -s /var/lib/pgsql/9.0/data -d /var/lib/pgsql/9.2 -u 9.2 -m
```

6. Start the PostgreSQL service, run the `analyze_new_cluster.sh` script and stop PostgreSQL.

```
/etc/init.d/postgresql-9.2 start
su - postgres -c "/var/lib/pgsql/9.2/analyze_new_cluster.sh"
```

```
/etc/init.d/postgresql-9.2 stop
```

7. [Click here](#) only if your reporting database is running on a separate port.

**Do this on the application server - my.app.host**

8. Move the collabnet repository of the older version of TeamForge.

```
mv /etc/yum.repos.d/collabnet-7.0.0.x.repo /etc/yum.repos.d/
collabnet-7.0.0.x.repo.cn_backup
```

 **Note:** Replace "x" with the appropriate patch release number if applicable.

9. Run the following commands to upgrade Red Hat/CentOS to the latest version.

- a) Remove the neon-devel package if you are upgrading from Teamforge 6.2.

```
yum erase neon-devel -y
yum erase subversion-devel -y
```

- b) Upgrade the operating system packages.

```
yum upgrade
```

10. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

- a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

- b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.

- c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```


See [Set up SELINUX](#) on page 271 to have TeamForge to run in SELinux mode after completing the installation or upgrade.

11. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

12. If the PostgreSQL database is running locally, stop the PostgreSQL service.


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

13. Uninstall the TeamForge CLI add-on (if it is already installed).

 **Note:** Skip this step if you are upgrading from TeamForge 6.2

```
cd /opt/collabnet/teamforge/add-ons/teamforge_cli_server
./install --uninstall
yum erase teamforge_cli_server
```

14. Uninstall the PostgreSQL RPMs.

 **Note:** When prompted, you must select the option to remove all the dependent packages.

```
yum erase postgresql90-libs postgresql90-docs postgresql90-server
postgresql90
```

15. Install the following application packages.

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

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


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

```
yum install teamforge-git
```

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

```
yum install teamforge-codesearch
```

16. In the `site-options.conf` file, make sure you do the following.

 **Note:** Back up your `site-options.conf` file before making any changes.

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

a) Update the host name and domain name, if required.

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

```
HOST_my.db.host=datamart etl
```

```
DOMAIN_localhost=my.app.domain.com
```

Configure the following settings if you are installing Git.

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

Configure the following settings if you are installing Black Duck Code Sight.

```
HOST_my.host.name=app database indexer subversion cvs codesearch
```

b) Configure the `JAVA_HOME` token for TeamForge.

```
JAVA_HOME=/usr/java/jdk1.7.0_40
```

c) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

- `SSL=on`
- `SSL_CERT_FILE=`
- `SSL_KEY_FILE=`
- `SSL_CA_CERT_FILE=`
- `SSL_CHAIN_FILE=`

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

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`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.

- f) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to `$auto$`, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- DATABASE\_PASSWORD
- DATABASE\_READ\_ONLY\_PASSWORD
- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD
- BDCS\_ADMIN\_PASSWORD
- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


g) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX=<A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`).

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- h) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and PostgreSQL automatically.
- i) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

- 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`

 **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 username add this
token:
BDCS_ADMIN_USERNAME=<sysadmin>
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) To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
- m) If you have LDAP set up for external authentication, you must set the **"REQUIRE\_USER\_PASSWORD\_CHANGE"** site options token to false.
- n) If you are installing TeamForge through disconnected media, set the token **HELP\_AVAILABILITY=local**.
- o) Ensure to set the token **SELINUX\_SETUP=false** temporarily in the `site-options.conf` file.
- p) Save the `site-options.conf` file.

17. Run the following command to remove the pagespeed cache.

```
cd /opt/collabnet/teamforge/cache
rm -rf pagespeed
```

18. Back up the old TeamForge runtime directory.


```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

19. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

20. 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 reporting server - my.reports.host**

21. Move the collabnet repository of the older version of TeamForge.

```
mv /etc/yum.repos.d/collabnet-7.0.0.x.repo /etc/yum.repos.d/
collabnet-7.0.0.x.repo.cn_backup
```

 **Note:** Replace "x" with the appropriate patch release number if applicable.

22. Run the following commands to upgrade Red Hat/CentOS to the latest version.

- a) Remove the neon-devel package if you are upgrading from Teamforge 6.2.

```
yum erase neon-devel -y
yum erase subversion-devel -y
```

- b) Upgrade the operating system packages.

```
yum upgrade
```

23. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

- a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

- b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.
- c) If not disabled, run the following command to disable SELinux.


```
setenforce 0
```

24. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

25. If the PostgreSQL database is running locally, stop the PostgreSQL service.

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

26. Uninstall the PostgreSQL RPMs.


 **Note:** When prompted, you must select the option to remove all the dependent packages.

```
yum erase postgresql90-libs postgresql90-docs postgresql90-server
postgresql90
```

27. Install the TeamForge database packages.

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

28. Copy the `site-options.conf` file from **my.app.host** and modify the token settings.

 **Note:** If you choose to use the old `site-options.conf` file, don't forget to copy the `AUTO_DATA` token from the application server.

```
HOST_localhost=datamart etl
```

```
DOMAIN_localhost=my.reports.domain.com
```

```
HOST_my.app.host=app database indexer subversion cvs
```

29. Back up the old TeamForge runtime directory.


```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

30. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

**Do this on the application server - my.app.host**

31. Convert your site data to work with TeamForge 7.1.


 **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 7.1 site.

32. Run the following script to upgrade the [index to Lucene 4.x format](#).

 **Note:** You must back up the existing search index directory before running this script. Refer to the `SEARCH_INDEX_LOCATION` token in the `runtime-options.conf` file to know the search index directory location.

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

33. Run the following script to upgrade the Subversion working copies.

```
/opt/collabnet/teamforge/runtime/scripts/svn-upgrade-working-copies.sh
```

**34. 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.

**Do this on the reporting server - my.reports.host****35. Start the ETL service.**

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

**Do the following on the application server - my.app.host****36. If you are upgrading from TeamForge 7.0, run the post-install.py script.**

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

**37. If you are upgrading from TeamForge 6.2 or earlier versions and the CLI reports are already configured, follow these steps:**

## a) Back up the CLI reports.

```
mkdir /root/backup
cd /root/backup
backup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> find . -name .svn | xargs rm -rf
```

## b) Remove the cli/reports folder from the branding repository.

**Note:** Do a SVN delete to remove this folder from the repository.

```
mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> svn delete reports
cli> svn commit -m "To delete the old CLI reports folder"
```

## c) Run the post-install.py script.

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

## d) Commit the backup 'pkg' folder which is available in this location branding/cli/custom-reports/

```
mkdir /root/restore
cd /root/restore
restore> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
mkdir custom-reports
cp -avx /root/backup/pkg custom-reports
svn add custom-reports
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>
```

When prompted, enter the user name (<admin\_username>) and password (<admin\_password>)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```

38. If you have installed Git, integrate gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

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
```

39. 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.

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

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

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

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL path of the SCM integration server, for example, "`http://myint.box.net/svn/repos`", where `myint.box` is the server with the SCM integration server.

In addition, if you add a new integration server at some point later, you must run this `svn_cache.sh` script, (after creating the new integration server), on the TeamForge application server.

41. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

**Do this on the reporting server - my.reports.host**

42. Run the following initial load jobs (ETL).

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 325.

43. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.1.


a) Log into your site as the administrator.

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


See [Customize anything on your site](#).

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

See [Create a site-wide broadcast](#).

 **Important:** Do not delete the `teamforge-installer/7.1.0.0` directory. You will need it for future maintenance and upgrades.

44. Remove the repository and the file system backup files from `/tmp/backup_dir` directory after the TeamForge site is up and running as expected.

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

To install TeamForge Orchestrate, see [TeamForge Orchestrate installation](#).

To upgrade Review Board [see these instructions](#).

To run TeamForge in SELINUX enabled mode, see [Set up SELINUX](#) on page 271

### Upgrade to TeamForge 7.1 - Black Duck Code Sight on a separate server


In this procedure, we'll assume that you are upgrading on the same server where your existing TeamForge site is running on 7.0. It is possible to simultaneously upgrade and move your site to a new hardware. However, since we are working with a dedicated installation, the priority here is to keep things as simple and quick as possible.

In this option, the following services run on the application server (we call this `my.app.host`).

- TeamForge Application Server
- ETL Server
- Database Server (Operational DB and Reports DB)
- GIT Integration Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer)

The following service runs on the Code Sight server. (We call this `my.codesight.host`)

- Code Sight server

 **Note:** For the ETL service to run as expected in a distributed TeamForge installation, all servers must have the same time zone.


**Log in to the server as root.**

**Do the following on the application server - `my.app.host`**


1. Stop TeamForge.

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

2. Migrate your PostgreSQL to the latest version supported by TeamForge 7.1.

 **Important:** You must have RHEL/CentOS 6.5 or later to run PGTurant. Upgrade the operating system packages (`yum upgrade`) before running PGTurant on sites running on RHEL/CentOS versions 6.4 or earlier.

- a) Install PGTurant. See [Upgrade PostgreSQL using PGTurant](#) on page 256 for more information.

 **Important:** If your TeamForge site has no internet access, contact the [CollabNet Support](#), get the `pgturant-8.0.0.0-x.noarch.rpm` package and unpack the RPM.

```
rpm -ivh pgturant-8.0.0.0-x.noarch.rpm
```

```
yum install pgturant -y
```

- b) Upgrade TeamForge PostgreSQL data directory to PostgreSQL 9.2.

```
cd /opt/collabnet/pgturant/bin/
```

```
./pgturant -s /var/lib/pgsql/9.0/data -d /var/lib/pgsql/9.2 -u 9.2 -m
```

3. Start the PostgreSQL service, run the `analyze_new_cluster.sh` script and stop PostgreSQL.


```
/etc/init.d/postgresql-9.2 start
```

```
su - postgres -c "/var/lib/pgsql/9.2/analyze_new_cluster.sh"
```

```
/etc/init.d/postgresql-9.2 stop
```

4. [Click here](#) only if your reporting database is running on a separate port.

5. Back up the file system data.

 **Tip:** `/tmp/backup_dir` is just an example. You can use any directory or partition you prefer to store your backup files.

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

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

```
cp -Rpfv /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 -Rpfv /gitroot /tmp/backup_dir
```

```
cp -Rpfv /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
backup_dir/gerrit
```

- b) Back up your SSH keys, if any.

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

6. Move the collabnet repository of the older version of TeamForge.

```
mv /etc/yum.repos.d/collabnet-7.0.0.x.repo /etc/yum.repos.d/
collabnet-7.0.0.x.repo.cn_backup
```

 **Note:** Replace "x" with the appropriate patch release number if applicable.

7. Run the following commands to upgrade Red Hat/CentOS to the latest version.

- a) Remove the neon-devel package if you are upgrading from Teamforge 6.2.

```
yum erase neon-devel -y
```

```
yum erase subversion-devel -y
```

- b) Upgrade the operating system packages.

```
yum upgrade
```

8. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

- a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

- b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.

- c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```


See [Set up SELINUX](#) on page 271 to have TeamForge to run in SELinux mode after completing the installation or upgrade.

9. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

10. If the PostgreSQL database is running locally, stop the PostgreSQL service.


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

11. Uninstall the TeamForge CLI add-on (if it is already installed).

-  **Note:** Skip this step if you are upgrading from TeamForge 6.2

```
cd /opt/collabnet/teamforge/add-ons/teamforge_cli_server
./install --uninstall
yum erase teamforge_cli_server
```

12. Uninstall the PostgreSQL RPMs.

-  **Note:** When prompted, you must select the option to remove all the dependent packages.

```
yum erase postgresql90-libs postgresql90-docs postgresql90-server
postgresql90
```

13. 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
```

14. In the `site-options.conf` file, make sure you do the following.

-  **Note:** Back up your `site-options.conf` file before making any changes.

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

- a) Update the host name and domain name, if required.

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

```
DOMAIN_localhost=my.app.domain.com
```

```
HOST_my.codesight.host=codesearch
```

Configure the following settings if you are installing Git.

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

- b) Configure the `JAVA_HOME` token for TeamForge.

```
JAVA_HOME=/usr/java/jdk1.7.0_40
```

- c) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

- `SSL=on`
- `SSL_CERT_FILE=`
- `SSL_KEY_FILE=`
- `SSL_CA_CERT_FILE=`
- `SSL_CHAIN_FILE=`

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- 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`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- f) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to `$auto$`, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- `DATABASE_PASSWORD`
- `DATABASE_READ_ONLY_PASSWORD`
- `REPORTS_DATABASE_PASSWORD`
- `REPORTS_DATABASE_READ_ONLY_PASSWORD`
- `ETL_SOAP_SHARED_SECRET`
- `JAMES_ADMIN_PASSWORD`
- `BDCS_ADMIN_PASSWORD`
- `MIRROR_DATABASE_PASSWORD` (applicable only if you are mirroring your database)


- g) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSSION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`:

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- h) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and PostgreSQL automatically.
- i) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

- 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.codesight.host>
```

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 username add this token:  
`BDCS_ADMIN_USERNAME=<sysadmin>`  
 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) To enable the history protection feature of TeamForge Git integration, set the `GERRIT_FORCE_HISTORY_PROTECTION=true`. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
- m) If you have LDAP set up for external authentication, you must set the `"REQUIRE_USER_PASSWORD_CHANGE"` site options token to false.

- n) If you are installing TeamForge through disconnected media, set the token `HELP_AVAILABILITY=local`.
- o) Ensure to set the token `SELINUX_SETUP=false` temporarily in the `site-options.conf` file.
- p) Save the `site-options.conf` file.

15. Run the following command to remove the pagespeed cache.

```
cd /opt/collabnet/teamforge/cache
rm -rf pagespeed
```

16. Back up the old TeamForge runtime directory.

```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

17. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

18. 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 my.codesight.host

19. Move the collabnet repository of the older version of TeamForge.

```
mv /etc/yum.repos.d/collabnet-7.0.0.x.repo /etc/yum.repos.d/
collabnet-7.0.0.x.repo.cn_backup
```



**Note:** Replace "x" with the appropriate patch release number if applicable.

20. Run the following commands to upgrade Red Hat/CentOS to the latest version.

- a) Remove the neon-devel package if you are upgrading from Teamforge 6.2.

```
yum erase neon-devel -y
yum erase subversion-devel -y
```

- b) Upgrade the operating system packages.

```
yum upgrade
```

21. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

- a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

- b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.
- c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```

22. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)


23. Stop the Black Duck Code Sight service.

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

24. Install Black Duck Code Sight.

```
yum install teamforge-codesearch
```

25. Copy the master `site-options.conf` file from `my.app.host` and modify the token settings.

-  **Note:** If you choose to use the old `site-options.conf` file, don't forget to copy the `AUTO_DATA` token from the application server.

```
HOST_localhost=codesearch
```

```
DOMAIN_localhost=my.codesight.domain.com
```

```
Host_my.app.host=app database datamart etl indexer subversion cvs
```

Save the `site-options.conf` file.

26. Back up the old TeamForge runtime directory.

```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

27. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

28. Run the following script to upgrade the Subversion working copies.

```
/opt/collabnet/teamforge/runtime/scripts/svn-upgrade-working-copies.sh
```


29. Start the Black Duck Code Sight service.

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

30. To install the license for Black Duck Code Sight follow [these instructions](#).

### Do this on my.app.host

31. Convert your site data to work with TeamForge 7.1.


-  **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 7.1 site.

32. Run the following script to upgrade the [index to Lucene 4.x format](#).

-  **Note:** You must back up the existing search index directory before running this script. Refer to the `SEARCH_INDEX_LOCATION` token in the `runtime-options.conf` file to know the search index directory location.

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

33. Run the following script to upgrade the Subversion working copies.

```
/opt/collabnet/teamforge/runtime/scripts/svn-upgrade-working-copies.sh
```

34. 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.

35. If you are upgrading from TeamForge 7.0, run the `post-install.py` script.




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

36. If you are upgrading from TeamForge 6.2 or earlier versions and the CLI reports are already configured, follow these steps:

a) Back up the CLI reports.

```
mkdir /root/backup
cd /root/backup
backup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> find . -name .svn | xargs rm -rf
```

b) Remove the cli/reports folder from the branding repository.

 **Note:** Do a SVN delete to remove this folder from the repository.

```
mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> svn delete reports
cli> svn commit -m "To delete the old CLI reports folder"
```

c) Run the post-install.py script.

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

d) Commit the backup 'pkg' folder which is available in this location branding/cli/custom-reports/

```
mkdir /root/restore
cd /root/restore
restore> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
mkdir custom-reports
cp -avx /root/backup/pkg custom-reports
svn add custom-reports
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>
```

When prompted, enter the user name (<admin\_username>) and password (<admin\_password>)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```

37. If you have installed Git, integrate gerrit by running the post-install.py script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.

- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

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
```


### Do this on my.codesight.host

**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.

```
/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:

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

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL path of the SCM integration server, for example, "`http://myint.box.net/svn/repos`", where `myint.box` is the server with the SCM integration server.

In addition, if you add a new integration server at some point later, you must run this `svn_cache.sh` script, (after creating the new integration server), on the TeamForge application server.

### Do this on my.app.host

**40.** If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

**41.** Run the following initial load jobs (ETL).

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 325.

**42.** Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.1.


a) Log into your site as the administrator.

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


See [Customize anything on your site](#).

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

See [Create a site-wide broadcast](#).

 **Important:** Do not delete the `teamforge-installer/7.1.0.0` directory. You will need it for future maintenance and upgrades.

43. Remove the repository and the file system backup files from `/tmp/backup_dir` directory after the TeamForge site is up and running as expected.

 **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](#).

To upgrade Review Board [see these instructions](#).

To run TeamForge in SELINUX enabled mode, see [Set up SELINUX](#) on page 271

### Upgrade to TeamForge 7.1 - GIT on a separate server


In this procedure, we'll assume that you are upgrading on the same server where your existing TeamForge site is running on 7.0. It is possible to simultaneously upgrade and move your site to a new hardware. However, since we are working with a dedicated installation, the priority here is to keep things as simple and quick as possible.

In this option, the following services run on the application server (we call this `my.app.host`).

- TeamForge Application Server
- Black Duck Code Sight Server
- ETL Server
- Database Server (Operational DB and Reports DB)
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer)

The following service runs on the GIT Integration Server. (We call this `my.git.host`)

- GIT Integration Server

 **Note:** For the ETL service to run as expected in a distributed TeamForge installation, all servers must have the same time zone.


### Log in to the server as root.

#### Do the following on the application server - `my.app.host`


1. Stop TeamForge.

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

2. Migrate your PostgreSQL to the latest version supported by TeamForge 7.1.

 **Important:** You must have RHEL/CentOS 6.5 or later to run PGTurant. Upgrade the operating system packages (`yum upgrade`) before running PGTurant on sites running on RHEL/CentOS versions 6.4 or earlier.

a) Install PGTurant. See [Upgrade PostgreSQL using PGTurant](#) on page 256 for more information.

 **Important:** If your TeamForge site has no internet access, contact the [CollabNet Support](#), get the `pgturant-8.0.0.0-x.noarch.rpm` package and unpack the RPM.

```
rpm -ivh pgturant-8.0.0.0-x.noarch.rpm
```

```
yum install pgturant -y
```

b) Upgrade TeamForge PostgreSQL data directory to PostgreSQL 9.2.


```
cd /opt/collabnet/pgturant/bin/
./pgturant -s /var/lib/pgsql/9.0/data -d /var/lib/pgsql/9.2 -u 9.2 -m
```

3. Start the PostgreSQL service, run the `analyze_new_cluster.sh` script and stop PostgreSQL.

```
/etc/init.d/postgresql-9.2 start
```

```
su - postgres -c "/var/lib/pgsql/9.2/analyze_new_cluster.sh"
/etc/init.d/postgresql-9.2 stop
```

4. [Click here](#) only if your reporting database is running on a separate port.
5. Back up the file system data.

 **Tip:** /tmp/backup\_dir is just an example. You can use any directory or partition you prefer to store your backup files.

- a) 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 -Rpfv /svnroot /sf-svnroot /cvsroot /opt/collabnet/teamforge/var /
tmp/backup_dir
```

- b) Back up your SSH keys, if any.
  - c) Back up your SSL certificates and keys, if any.
6. Move the collabnet repository of the older version of TeamForge.

```
mv /etc/yum.repos.d/collabnet-7.0.0.x.repo /etc/yum.repos.d/
collabnet-7.0.0.x.repo.cn_backup
```

 **Note:** Replace "x" with the appropriate patch release number if applicable.

7. Run the following commands to upgrade Red Hat/CentOS to the latest version.
  - a) Remove the neon-devel package if you are upgrading from Teamforge 6.2.

```
yum erase neon-devel -y
yum erase subversion-devel -y
```

- b) Upgrade the operating system packages.

```
yum upgrade
```

8. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.
  - a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

- b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.
- c) If not disabled, run the following command to disable SELinux.


```
setenforce 0
```

See [Set up SELINUX](#) on page 271 to have TeamForge to run in SELinux mode after completing the installation or upgrade.

9. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)
10. If the PostgreSQL database is running locally, stop the PostgreSQL service.


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

11. Uninstall the TeamForge CLI add-on (if it is already installed).

 **Note:** Skip this step if you are upgrading from TeamForge 6.2

```
cd /opt/collabnet/teamforge/add-ons/teamforge_cli_server
./install --uninstall
yum erase teamforge_cli_server
```

## 12. Uninstall the PostgreSQL RPMs.

 **Note:** When prompted, you must select the option to remove all the dependent packages.

```
yum erase postgresql90-libs postgresql90-docs postgresql90-server
postgresql90
```

## 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 the Black Duck Code Sight packages run the following command:

```
yum install teamforge-codesearch
```

## 14. In the `site-options.conf` file, make sure you do the following.

 **Note:** Back up your `site-options.conf` file before making any changes.

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

a) Update the host name and domain name, if required.

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

```
DOMAIN_localhost=my.app.domain.com
```

```
HOST_my.git.host=gerrit
```

Configure the following settings if you are installing Black Duck Code Sight.

```
HOST_my.localhost= app database datamart etl indexer subversion
cvs codesearch
```

b) Configure the `JAVA_HOME` token for TeamForge.

```
JAVA_HOME=/usr/java/jdk1.7.0_40
```

c) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

- `SSL=on`
- `SSL_CERT_FILE=`
- `SSL_KEY_FILE=`
- `SSL_CA_CERT_FILE=`
- `SSL_CHAIN_FILE=`

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- 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`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- f) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to `$auto$`, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- DATABASE\_PASSWORD
- DATABASE\_READ\_ONLY\_PASSWORD
- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD
- BDCS\_ADMIN\_PASSWORD
- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


g) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any Alphanumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF};`.

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- h) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and PostgreSQL automatically.
- i) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

```
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
```

- 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
```

-  **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 username add this token:

```
BDCS_ADMIN_USERNAME=<sysadmin>
```

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) To enable the history protection feature of TeamForge Git integration, set the **`GERRIT_FORCE_HISTORY_PROTECTION=true`**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
- m) If you have LDAP set up for external authentication, you must set the **`REQUIRE_USER_PASSWORD_CHANGE`** site options token to false.
- n) If you are installing TeamForge through disconnected media, set the token **`HELP_AVAILABILITY=local`**.
- o) Ensure to set the token **`SELINUX_SETUP=false`** temporarily in the `site-options.conf` file.
- p) Save the `site-options.conf` file.

15. Run the following command to remove the pagespeed cache.

```
cd /opt/collabnet/teamforge/cache
```

```
rm -rf pagespeed
```

16. Back up the old TeamForge runtime directory.

```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```


17. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
```


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

18. 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.

19. Convert your site data to work with TeamForge 7.1.


-  **Tip:** Before you kick off the data migration, use the `/etc/init.d/collabnet status` command to make sure the Jboss, Tomcat and Tomcats 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 7.1 site.

20. Run the following script to upgrade the *index to Lucene 4.x format*.

-  **Note:** You must back up the existing search index directory before running this script. Refer to the `SEARCH_INDEX_LOCATION` token in the `runtime-options.conf` file to know the search index directory location.

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

21. Run the following script to upgrade the Subversion working copies.

```
/opt/collabnet/teamforge/runtime/scripts/svn-upgrade-working-copies.sh
```

22. 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.

23. If you are upgrading from TeamForge 7.0, run the `post-install.py` script.


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

24. If you are upgrading from TeamForge 6.2 or earlier versions and the CLI reports are already configured, follow these steps:

- a) Back up the CLI reports.

```
mkdir /root/backup
cd /root/backup
backup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> find . -name .svn | xargs rm -rf
```

- b) Remove the `cli/reports` folder from the branding repository.

-  **Note:** Do a SVN delete to remove this folder from the repository.

```
mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> svn delete reports
cli> svn commit -m "To delete the old CLI reports folder"
```

- c) Run the `post-install.py` script.

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

- d) Commit the backup 'pkg' folder which is available in this location `branding/cli/custom-reports/`



```

mkdir /root/restore
cd /root/restore
restore> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
mkdir custom-reports
cp -avx /root/backup/pkg custom-reports
svn add custom-reports
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>

```


When prompted, enter the user name (<admin\_username>) and password (<admin\_password>)

```

ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q

```

25. Install the Black Duck Code Sight license on the server where Black Duck Code Sight is installed. For more information see [these instructions](#).
26. 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.

```

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

```

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

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

```

cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>

```

Provide a repository base URL path of the SCM integration server, for example, "`http://myint.box.net/svn/repos`", where `myint.box` is the server with the SCM integration server.

In addition, if you add a new integration server at some point later, you must run this `svn_cache.sh` script, (after creating the new integration server), on the TeamForge application server.

28. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```

cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss

```

29. Run the following initial load jobs (ETL).

- 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 325.

### Do this on the Git integration server - my.git.host

30. Migrate your PostgreSQL to the latest version supported by TeamForge 7.1.

 **Important:** You must have RHEL/CentOS 6.5 or later to run PGTurant. Upgrade the operating system packages (`yum upgrade`) before running PGTurant on sites running on RHEL/CentOS versions 6.4 or earlier.

a) Install PGTurant. See [Upgrade PostgreSQL using PGTurant](#) on page 256 for more information.

 **Important:** If your TeamForge site has no internet access, contact the [CollabNet Support](#), get the `pgturant-8.0.0.0-x.noarch.rpm` package and unpack the RPM.

```
rpm -ivh pgturant-8.0.0.0-x.noarch.rpm
```

```
yum install pgturant -y
```

b) Upgrade TeamForge PostgreSQL data directory to PostgreSQL 9.2.


```
cd /opt/collabnet/pgturant/bin/
./pgturant -s /var/lib/pgsql/9.0/data -d /var/lib/pgsql/9.2 -u 9.2 -m
```

31. Start the PostgreSQL service, run the `analyze_new_cluster.sh` script and stop PostgreSQL.

```
/etc/init.d/postgresql-9.2 start
su - postgres -c "/var/lib/pgsql/9.2/analyze_new_cluster.sh"
/etc/init.d/postgresql-9.2 stop
```

32. Back up the Git file system data.

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

 **Tip:** `/tmp` is just an example. You can use any directory or partition that you prefer.

| Directory             | Contents                     |
|-----------------------|------------------------------|
| <code>/gitroot</code> | Git source code repositories |

```
cp -Rpfv /gitroot /tmp/gitbackup_dir
cp -Rpfv /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
gitbackup_dir/gerrit
```

b) Back up your SSH keys, if any.

33. Move the collabnet repository of the older version of TeamForge.

```
mv /etc/yum.repos.d/collabnet-7.0.0.x.repo /etc/yum.repos.d/
collabnet-7.0.0.x.repo.cn_backup
```

 **Note:** Replace "x" with the appropriate patch release number if applicable.

34. Run the following commands to upgrade Red Hat/CentOS to the latest version.

a) Remove the neon-devel package if you are upgrading from Teamforge 6.2.

```
yum erase neon-devel -y
yum erase subversion-devel -y
```

b) Upgrade the operating system packages.

```
yum upgrade
```

35. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.

- c) If not disabled, run the following command to disable SELinux.


```
setenforce 0
```

36. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

37. Install the Git packages.

```
yum install teamforge-git
```

38. Copy the `site-options.conf` file from `my.app.host` and modify the token settings.

-  **Note:** If you choose to use the old `site-options.conf` file, don't forget to copy the `AUTO_DATA` token from the application server.

```
HOST_localhost=gerrit
```

```
DOMAIN_localhost=my.git.domain.com
```

```
HOST_my.app.host=app database datamart etl indexer subversion cvs
```

39. Back up the old TeamForge runtime directory.

```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

40. Run the installer.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

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

```
/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:

Login to the app server and run the following command:

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

42. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.1.


- a) Log into your site as the administrator.

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

See [Customize anything on your site](#).

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

See [Create a site-wide broadcast](#).

-  **Important:** Do not delete the `teamforge-installer/7.1.0.0` directory. You will need it for future maintenance and upgrades.

43. Remove the repository and the file system backup files from `/tmp/backup_dir` directory after the TeamForge site is up and running as expected.

- 👉 **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](#).

To upgrade Review Board [see these instructions](#).

To run TeamForge in SELINUX enabled mode, see [Set up SELINUX](#) on page 271

### Upgrade to TeamForge 7.1 on new hardware - All Services on the same server

To upgrade to TeamForge 7.1, set up a new hardware, then bring your old site's data and convert it.

In this option, the following services run on the application server (we call this my.app.host).

- TeamForge Application Server
- Black Duck Code Sight Server
- Database Server (Operational DB and Reports DB)
- ETL Server
- GIT Integration Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer).

### Log in to the server as root.

### Do the following on the existing TeamForge application server - my.app.host

1. Stop TeamForge.

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

2. Migrate your PostgreSQL to the latest version supported by TeamForge 7.1.

- 👉 **Important:** You must have RHEL/CentOS 6.5 or later to run PGTurant. Upgrade the operating system packages (`yum upgrade`) before running PGTurant on sites running on RHEL/CentOS versions 6.4 or earlier.

- a) Install PGTurant. See [Upgrade PostgreSQL using PGTurant](#) on page 256 for more information.

- 👉 **Important:** If your TeamForge site has no internet access, contact the [CollabNet Support](#), get the `pgturant-8.0.0.0-x.noarch.rpm` package and unpack the RPM.

```
rpm -ivh pgturant-8.0.0.0-x.noarch.rpm
```

```
yum install pgturant -y
```

- b) Upgrade TeamForge PostgreSQL data directory to PostgreSQL 9.2.

```
cd /opt/collabnet/pgturant/bin/
```

```
./pgturant -s /var/lib/pgsql/9.0/data -d /var/lib/pgsql/9.2 -u 9.2 -m
```

3. Start the PostgreSQL service, run the `analyze_new_cluster.sh` script and stop PostgreSQL.

```
/etc/init.d/postgresql-9.2 start
```

```
su - postgres -c "/var/lib/pgsql/9.2/analyze_new_cluster.sh"
```

```
/etc/init.d/postgresql-9.2 stop
```

4. [Click here](#) only if your reporting database is running on a separate port.

5. Back up your site database.

- 👉 **Tip:** `/tmp/backup_dir` is just an example. You can use any directory or partition you prefer to store your backup files.


```
mkdir -p /tmp/backup_dir
```

```
cd /var/lib
```


```
tar -zcvf /tmp/backup_dir/pgsql.tgz pgsql/9.2
```

6. If you have Black Duck Code Sight installed, then back up the Black Duck Code Sight data.

```
cd /opt/collabnet
tar czvf /tmp/backup_dir/blackduck.tgz blackduck
```

-  **Tip:** If the Black Duck Code Sight directory size is huge, the back up task may run for longer duration. You may proceed with the following steps while the Black Duck Code Sight is being backed up.

7. Back up the file system data.

-  **Tip:** /tmp/backup\_dir is just an example. You can use any directory or partition you prefer to store your backup files.

- a) 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 -Rpfv /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 -Rpfv /gitroot /tmp/backup_dir
cp -Rpfv /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
backup_dir/gerrit
```


Compress your backup data.

```
cd /tmp
tar czvf 70backup.tgz backup_dir
```

- b) Back up your SSH keys, if any.  
 c) Back up your SSL certificates and keys, if any.
8. Copy the master configuration file from the old server to the same location on the new server.

```
scp /opt/collabnet/teamforge-installer/7.0.0.x/conf/site-options.conf
username@newbox:/tmp
```

-  **Note:** Replace "x" with the appropriate patch release number if applicable.

-  **Tip:** scp is just an example. You can choose any file transfer method you prefer.

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

```
scp /tmp/70backup.tgz username@newbox:/tmp
```

#### Do the following on the new TeamForge Application Server

10. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

- a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

- b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.  
 c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```

See [Set up SELINUX](#) on page 271 to have TeamForge to run in SELinux mode after completing the installation or upgrade.

**11.** Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

**12.** 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 the Black Duck Code Sight packages run the following command:

```
yum install teamforge-codesearch
```

**13.** Reload the PostgreSQL data.

```
cd /var/lib
mv pgsql pgsql_orig
tar -zxvf /tmp/backup_dir/pgsql.tgz
```


**14.** Copy the `site-options.conf` file to the TeamForge installer directory.

```
cp /tmp/site-options.conf /opt/collabnet/teamforge-installer/7.1.0.0/conf
```

**15.** Unpack the file system data.

```
cd /tmp
tar xzvf 70backup.tgz
```

**16.** In the `site-options.conf` file, make sure you do the following.

 **Note:** Back up your `site-options.conf` file before making any changes.

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

a) Update the host name and domain name, if required.

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

```
DOMAIN_localhost=my.app.domain.com
```

Configure the following settings if you are installing Git.

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

Configure the following settings if you are installing Black Duck Code Sight.

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

b) Configure the `JAVA_HOME` token for TeamForge.

```
JAVA_HOME=/usr/java/jdk1.7.0_40
```

- c) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

- `SSL=on`
- `SSL_CERT_FILE=`
- `SSL_KEY_FILE=`
- `SSL_CA_CERT_FILE=`
- `SSL_CHAIN_FILE=`

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- 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`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- f) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to `$auto$`, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- `DATABASE_PASSWORD`
- `DATABASE_READ_ONLY_PASSWORD`
- `REPORTS_DATABASE_PASSWORD`
- `REPORTS_DATABASE_READ_ONLY_PASSWORD`
- `ETL_SOAP_SHARED_SECRET`
- `JAMES_ADMIN_PASSWORD`
- `BDCS_ADMIN_PASSWORD`
- `MIRROR_DATABASE_PASSWORD` (applicable only if you are mirroring your database)


- g) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX=<A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`.

 **Important:** The password-related tokens cannot contain the following characters: \$<>/\ ' " ` in the `site-options.conf` file.

- h) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and Postgresql automatically.
- i) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

- 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.app.host or my.domain.com>
```

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 username add this token:

```
BDCS_ADMIN_USERNAME=<sysadmin>
```

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) To enable the history protection feature of TeamForge Git integration, set the `GERRIT_FORCE_HISTORY_PROTECTION=true`. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
- m) If you have LDAP set up for external authentication, you must set the `"REQUIRE_USER_PASSWORD_CHANGE"` site options token to false.
- n) If you are installing TeamForge through disconnected media, set the token `HELP_AVAILABILITY=local`.
- o) Ensure to set the token `SELINUX_SETUP=false` temporarily in the `site-options.conf` file.
- p) Save the `site-options.conf` file.

## 17. Recreate the runtime environment.



```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

**18. Reload the svnroot, sf-svnroot, cvsroot, gitroot and var directories.**

```
cp -Rpfv /tmp/backup_dir/svnroot /svnroot
cp -Rpfv /tmp/backup_dir/cvsroot /cvsroot
cp -Rpfv /tmp/backup_dir/sf-svnroot /sf-svnroot
cp -Rpfv /tmp/backup_dir/var /opt/collabnet/teamforge/var
```

If Git integration is enabled, do the following:


```
cp -Rpfv /tmp/backup_dir/gitroot /
cp -Rpfv /tmp/backup_dir/gerrit/etc /opt/collabnet/gerrit
cp -Rpf /tmp/backup_dir/gerrit/.ssh /opt/collabnet/gerrit
```

**19. Recreate the runtime environment to set the database credentials.**


```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

**20. 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.

**21. Convert your site data to work with TeamForge 7.1.**


 **Tip:** Before you kick off the data migration, use the `/etc/init.d/collabnet status` command to make sure the Jboss, Tomcat and Tomcats 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 7.1 site.

**22. Run the following script to upgrade the [index to Lucene 4.x format](#).**

 **Note:** You must back up the existing search index directory before running this script. Refer to the `SEARCH_INDEX_LOCATION` token in the `runtime-options.conf` file to know the search index directory location.

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

**23. Run the following script to upgrade the Subversion working copies.**

```
/opt/collabnet/teamforge/runtime/scripts/svn-upgrade-working-copies.sh
```

**24. 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.

**25. If you are upgrading from TeamForge 7.0, run the `post-install.py` script.**


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

26. If you are upgrading from TeamForge 6.2 or earlier versions and the CLI reports are already configured, follow these steps:

a) Back up the CLI reports.

```
mkdir /root/backup
cd /root/backup
backup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> find . -name .svn | xargs rm -rf
```

b) Remove the cli/reports folder from the branding repository.

 **Note:** Do a SVN delete to remove this folder from the repository.

```
mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> svn delete reports
cli> svn commit -m "To delete the old CLI reports folder"
```

c) Run the post-install.py script.

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

d) Commit the backup 'pkg' folder which is available in this location branding/cli/custom-reports/

```
mkdir /root/restore
cd /root/restore
restore> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
mkdir custom-reports
cp -avx /root/backup/pkg custom-reports
svn add custom-reports
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>
```

When prompted, enter the user name (<admin\_username>) and password (<admin\_password>)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```

27. If you have installed Git, integrate gerrit by running the post-install.py script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.

- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

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
```


**28.** Install the Black Duck Code Sight license on the server where Black Duck Code Sight is installed. For more information see [these instructions](#).

**29.** 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.

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

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

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

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL path of the SCM integration server, for example, "`http://myint.box.net/svn/repos`", where `myint.box` is the server with the SCM integration server.

In addition, if you add a new integration server at some point later, you must run this `svn_cache.sh` script, (after creating the new integration server), on the TeamForge application server.

**31.** If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

**32.** Run the following initial load jobs (ETL).

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 325.

**33.** Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.1.

a) Log into your site as the administrator.

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

See [Customize anything on your site](#).

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

See [Create a site-wide broadcast](#).

- 👉 **Important:** Do not delete the `teamforge-installer/7.1.0.0` directory. You will need it for future maintenance and upgrades.

34. Remove the repository and the file system backup files from `/tmp/backup_dir` directory after the TeamForge site is up and running as expected.

- 👉 **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](#).

To upgrade Review Board [see these instructions](#).

To run TeamForge in SELINUX enabled mode, see [Set up SELINUX](#) on page 271

### Upgrade to TeamForge 7.1 - Database and SCM on separate servers

In this procedure, we'll assume that you are upgrading on the same server where your existing TeamForge 7.0 site is running. It is possible to simultaneously upgrade and move your site to a new hardware. However, since we are working with a dedicated installation, the priority here is to keep things as simple and quick as possible.

In this option, the following services run on the application server (we call this `my.app.host`).

- TeamForge Application Server
- Black Duck Code Sight Server
- ETL Server
- Search Server (Indexer).

The following service runs on the database server (We call this `my.db.host`)

- Database Server (Operational DB and Reports DB)

The following services run of the SCM server (We call this `my.scm.host`)

- SCM Integration Server (Subversion and CVS)
- GIT Integration Server

- 👉 **Note:** For the ETL service to run as expected in a distributed TeamForge installation, all servers must have the same time zone.

### Log in to the server as root.

#### Do the following on the application server - `my.app.host`

1. Stop TeamForge.

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

2. Back up the file system data.

- 👉 **Tip:** `/tmp/backup_dir` is just an example. You can use any directory or partition you prefer to store your backup files.

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

| Directory                                 | Contents                                        |
|-------------------------------------------|-------------------------------------------------|
| <code>/opt/collabnet/teamforge/var</code> | User-created data, such as artifact attachments |
| <code>/sf-svnroot</code>                  | Subversion repository for branding data         |


```
mkdir -p /tmp/backup_dir
```

```
cp -Rpfv /sf-svnroot /opt/collabnet/teamforge/var /tmp/backup_dir
```


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

**Do this on the database server - my.db.host**

3. Migrate your PostgreSQL to the latest version supported by TeamForge 7.1.

 **Important:** You must have RHEL/CentOS 6.5 or later to run PGTurant. Upgrade the operating system packages (`yum upgrade`) before running PGTurant on sites running on RHEL/CentOS versions 6.4 or earlier.

- a) Install PGTurant. See [Upgrade PostgreSQL using PGTurant](#) on page 256 for more information.

 **Important:** If your TeamForge site has no internet access, contact the [CollabNet Support](#), get the `pgturant-8.0.0.0-x.noarch.rpm` package and unpack the RPM.

```
rpm -ivh pgturant-8.0.0.0-x.noarch.rpm
yum install pgturant -y
```

- b) Upgrade TeamForge PostgreSQL data directory to PostgreSQL 9.2.

```
cd /opt/collabnet/pgturant/bin/
./pgturant -s /var/lib/pgsql/9.0/data -d /var/lib/pgsql/9.2 -u 9.2 -m
```

4. Start the PostgreSQL service, run the `analyze_new_cluster.sh` script and stop PostgreSQL.

```
/etc/init.d/postgresql-9.2 start
su - postgres -c "/var/lib/pgsql/9.2/analyze_new_cluster.sh"
/etc/init.d/postgresql-9.2 stop
```

5. [Click here](#) only if your reporting database is running on a separate port.

**Do this on the application server - my.app.host**

6. Move the collabnet repository of the older version of TeamForge.

```
mv /etc/yum.repos.d/collabnet-7.0.0.x.repo /etc/yum.repos.d/
collabnet-7.0.0.x.repo.cn_backup
```

 **Note:** Replace "x" with the appropriate patch release number if applicable.

7. Run the following commands to upgrade Red Hat/CentOS to the latest version.

- a) Remove the neon-devel package if you are upgrading from Teamforge 6.2.

```
yum erase neon-devel -y
yum erase subversion-devel -y
```

- b) Upgrade the operating system packages.

```
yum upgrade
```

8. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

- a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

- b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.


- c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```

See [Set up SELINUX](#) on page 271 to have TeamForge to run in SELinux mode after completing the installation or upgrade.

9. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

10. Uninstall the TeamForge CLI add-on (if it is already installed).

 **Note:** Skip this step if you are upgrading from TeamForge 6.2

```
cd /opt/collabnet/teamforge/add-ons/teamforge_cli_server
./install --uninstall
```

```
yum erase teamforge_cli_server
```

**11. Install the following application packages.**


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

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

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

```
yum install teamforge-codesearch
```

**12. In the `site-options.conf` file, make sure you do the following.**

-  **Note:** Back up your `site-options.conf` file before making any changes.

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

- a) Update the host name and domain name, if required.

```
HOST_localhost=app etl indexer
```

```
HOST_my.db.host=database datamart
```

```
HOST_my.scm.host=subversion cvs Gerrit
```

```
DOMAIN_localhost=my.app.domain.com
```

Configure the following settings if you are installing Black Duck Code Sight.

```
HOST_my.localhost=app etl indexer codesearch
```

- b) Configure the `JAVA_HOME` token for TeamForge.

```
JAVA_HOME=/usr/java/jdk1.7.0_40
```

- c) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

- `SSL=on`
- `SSL_CERT_FILE=`
- `SSL_KEY_FILE=`
- `SSL_CA_CERT_FILE=`
- `SSL_CHAIN_FILE=`

-  **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- 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`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- f) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to `$auto$`, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- `DATABASE_PASSWORD`
- `DATABASE_READ_ONLY_PASSWORD`
- `REPORTS_DATABASE_PASSWORD`
- `REPORTS_DATABASE_READ_ONLY_PASSWORD`
- `ETL_SOAP_SHARED_SECRET`
- `JAMES_ADMIN_PASSWORD`
- `BDCS_ADMIN_PASSWORD`
- `MIRROR_DATABASE_PASSWORD` (applicable only if you are mirroring your database)


g) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`.

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- h) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and PostgreSQL automatically.
- i) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

- 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
```

-  **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 username add this
token:
BDCS_ADMIN_USERNAME=<sysadmin>
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) To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
  - m) If you have LDAP set up for external authentication, you must set the **"REQUIRE\_USER\_PASSWORD\_CHANGE"** site options token to false.
  - n) If you are installing TeamForge through disconnected media, set the token **HELP\_AVAILABILITY=local**.
  - o) Ensure to set the token **SELINUX\_SETUP=false** temporarily in the `site-options.conf` file.
  - p) Save the `site-options.conf` file.
- 13.** Run the following command to remove the pagespeed cache.

```
cd /opt/collabnet/teamforge/cache
rm -rf pagespeed
```

- 14.** Back up the old TeamForge runtime directory.


```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

- 15.** Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

- 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 the following on the database server - my.db.host

- 17.** Move the collabnet repository of the older version of TeamForge.

```
mv /etc/yum.repos.d/collabnet-7.0.0.x.repo /etc/yum.repos.d/
collabnet-7.0.0.x.repo.cn_backup
```

-  **Note:** Replace "x" with the appropriate patch release number if applicable.

- 18.** Run the following commands to upgrade Red Hat/CentOS to the latest version.

- a) Remove the neon-devel package if you are upgrading from Teamforge 6.2.

```
yum erase neon-devel -y
```



```
yum erase subversion-devel -y
```

- b) Upgrade the operating system packages.

```
yum upgrade
```

19. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

- a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

- b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.  
c) If not disabled, run the following command to disable SELinux.


```
setenforce 0
```

20. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

21. If the PostgreSQL database is running locally, stop the PostgreSQL service.

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

22. Uninstall the PostgreSQL RPMs.


-  **Note:** When prompted, you must select the option to remove all the dependent packages.

```
yum erase postgresql90-libs postgresql90-docs postgresql90-server
postgresql90
```

23. Install the TeamForge database packages.

```
yum install teamforge-database
```

24. Copy the `site-options.conf` file from `my.app.host` and modify the token settings.

-  **Note:** If you choose to use the old `site-options.conf` file, don't forget to copy the `AUTO_DATA` token from the application server.

```
HOST_localhost=database datamart
```

```
DOMAIN_localhost=my.db.domain.com
```

```
HOST_my.app.host=app etl indexer
```

```
HOST_my.scm.host=subversion cvs gerrit
```

25. Back up the old TeamForge runtime directory.


```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

26. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

### Do this on application server - my.app.host

27. Convert your site data to work with TeamForge 7.1.


-  **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 7.1 site.

28. Run the following script to upgrade the *index to Lucene 4.x format*.

-  **Note:** You must back up the existing search index directory before running this script. Refer to the `SEARCH_INDEX_LOCATION` token in the `runtime-options.conf` file to know the search index directory location.

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

29. Run the following script to upgrade the Subversion working copies.

```
/opt/collabnet/teamforge/runtime/scripts/svn-upgrade-working-copies.sh
```

30. 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.

31. If you are upgrading from TeamForge 7.0, run the `post-install.py` script.


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

32. If you are upgrading from TeamForge 6.2 or earlier versions and the CLI reports are already configured, follow these steps:

a) Back up the CLI reports.

```
mkdir /root/backup
cd /root/backup
backup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> find . -name .svn | xargs rm -rf
```

b) Remove the `cli/reports` folder from the branding repository.

-  **Note:** Do a SVN delete to remove this folder from the repository.

```
mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> svn delete reports
cli> svn commit -m "To delete the old CLI reports folder"
```

c) Run the `post-install.py` script.

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

d) Commit the backup 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```
mkdir /root/restore
cd /root/restore
restore> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
mkdir custom-reports
cp -avx /root/backup/pkg custom-reports
svn add custom-reports
```

```
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>
```

When prompted, enter the user name (<admin\_username>) and password (<admin\_password>)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```


33. Install the Black Duck Code Sight license on the server where Black Duck Code Sight is installed. For more information see [these instructions](#).

34. 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.

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

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

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

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL path of the SCM integration server, for example, "`http://myint.box.net/svn/repos`", where `myint.box` is the server with the SCM integration server.

In addition, if you add a new integration server at some point later, you must run this `svn_cache.sh` script, (after creating the new integration server), on the TeamForge application server.

36. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

37. Run the following initial load jobs (ETL).

- 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 325.

### Do the following on the SCM server - `my.scm.host`


If your TeamForge setup includes source control running on its own server, you'll have to upgrade that server as well as the main TeamForge application server.

38. Stop TeamForge.


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

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

### 39. Migrate your PostgreSQL to the latest version supported by TeamForge 7.1.

-  **Important:** You must have RHEL/CentOS 6.5 or later to run PGTurant. Upgrade the operating system packages (`yum upgrade`) before running PGTurant on sites running on RHEL/CentOS versions 6.4 or earlier.

- a) Install PGTurant. See [Upgrade PostgreSQL using PGTurant](#) on page 256 for more information.

-  **Important:** If your TeamForge site has no internet access, contact the [CollabNet Support](#), get the `pgturant-8.0.0.0-noarch.rpm` package and unpack the RPM.

```
rpm -ivh pgturant-8.0.0.0-x.noarch.rpm
```

```
yum install pgturant -y
```


- b) Upgrade TeamForge PostgreSQL data directory to PostgreSQL 9.2.

```
cd /opt/collabnet/pgturant/bin/
./pgturant -s /var/lib/pgsql/9.0/data -d /var/lib/pgsql/9.2 -u 9.2 -m
```

### 40. Start the PostgreSQL service, run the `analyze_new_cluster.sh` script and stop PostgreSQL.

```
/etc/init.d/postgresql-9.2 start
su - postgres -c "/var/lib/pgsql/9.2/analyze_new_cluster.sh"
/etc/init.d/postgresql-9.2 stop
```

### 41. Back up your SCM data.

-  **Tip:** `/tmp` in the following step is just an example. You can use any directory or partition that you prefer.

- a) Make an archive with the following data directories.

| Directory             | Contents                                                |
|-----------------------|---------------------------------------------------------|
| <code>/svnroot</code> | Subversion source code repositories                     |
| <code>/cvsroot</code> | CVS source code repositories (not present in all sites) |
| <code>/gitroot</code> | Git source code repositories                            |

```
mkdir -p /tmp/scmbackup_dir/gerrit
cp -Rpfv /svnroot /cvsroot /tmp/scmbackup_dir
```

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

### 42. Move the collabnet repository of the older version of TeamForge.

```
mv /etc/yum.repos.d/collabnet-7.0.0.x.repo /etc/yum.repos.d/
collabnet-7.0.0.x.repo.cn_backup
```

-  **Note:** Replace "x" with the appropriate patch release number if applicable.

### 43. Run the following commands to upgrade Red Hat/CentOS to the latest version.

- a) Remove the `neon-devel` package if you are upgrading from Teamforge 6.2.

```
yum erase neon-devel -y
yum erase subversion-devel -y
```

- b) Upgrade the operating system packages.

```
yum upgrade
```

### 44. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

- a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

- b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.
- c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```


45. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

46. If Git is enabled and the PostgreSQL database is running locally, follow these steps.

- a) Stop the PostgreSQL service.

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

- b) Uninstall the PostgreSQL RPMs.

 **Note:** When prompted, you must select the option to remove all the dependent packages.

```
yum erase postgresql90-libs postgresql90-docs postgresql90-server
postgresql90
```

47. Install the following application packages.


- a) Install the source code component.

```
yum install teamforge-scm
```

- b) To install the Git packages, run the following command.

```
yum install teamforge-git
```

48. Copy the master `site-options.conf` file from `my.app.host` and modify the host token settings in the `site-options.conf` file.

 **Note:** If you choose to use the old `site-options.conf` file, don't forget to copy the `AUTO_DATA` token from the application server.

```
HOST_localhost=subversion cvs gerrit
```

```
DOMAIN_localhost=my.scm.domain.com
```

```
HOST_my.app.host=app etl indexer codesearch
```

```
HOST_my.db.host=database datamart
```

Save the `site-options.conf` file.

49. Back up the old TeamForge runtime directory.


```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

50. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

51. 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.

52. Start the Tomcat service.

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

53. If you have installed Git, integrate gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
  - TeamForge password: The password for the dedicated TeamForge site administrator account.
  - Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.
- 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
```

54. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.1.

- a) Log into your site as the administrator.
- b) If your site has custom branding, verify that your branding changes still work as intended.  
See [Customize anything on your site](#).
- c) Let your site's users know they've been upgraded.  
See [Create a site-wide broadcast](#).

 **Important:** Do not delete the `teamforge-installer/7.1.0.0` directory. You will need it for future maintenance and upgrades.

55. Remove the repository and the file system backup files from `/tmp/backup_dir` directory after the TeamForge site is up and running as expected.

 **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](#).

To upgrade Review Board [see these instructions](#).

To run TeamForge in SELINUX enabled mode, see [Set up SELINUX](#) on page 271

### Upgrade an advanced TeamForge site to TeamForge 7.1

Upgrading to TeamForge 7.1 on an advanced site can be complicated but you get more flexibility and control.

If there is any doubt about what kind of site you are working with, see [Is my TeamForge site "dedicate"d or "advanced"?](#) on page 253

### Upgrade to TeamForge 7.1 with Oracle Database services on a separate server

In this procedure, we'll assume that you are upgrading on the same server where your existing TeamForge 7.0 site is running.

In this option, the following services run on the TeamForge Application Server (We call this `my.app.host`).


- TeamForge Application Server
- Black Duck Code Sight Server
- ETL Server
- Search Server (Indexer)

The following service runs on the Database server (Oracle). (We call this `my.db.host`)

- Database Server (Operational DB and Reports DB)

The following service run on the SCM server.(We call this `my.scn.host`)

- SCM Integration Server (Subversion and CVS)

 **Note:** For the ETL service to run as expected in a distributed TeamForge installation, all servers must have the same time zone.

### Log in to the servers as root.

#### Do this on the Oracle database server - my.db.host

1. Make a dump file of your site database.


To back up the Oracle database, follow the [Oracle backup procedure](#).

#### Do the following on the TeamForge Application Server (We call this my.app.host)

2. Stop the Apache server and the TeamForge application server.

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

3. Back up the file system data.


 **Tip:** /tmp in the following step is just an example. You can use any directory or partition that you prefer.

- a) Make an archive 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 -p /tmp/backup_dir
cp -Rpfv /sf-svnroot /opt/collabnet/teamforge/var /tmp/backup_dir
```

4. If the SCM services are running on the TeamForge Application Server (my.app.host), do the following.

 **Tip:** /tmp in the following step is just an example. You can use any directory or partition that you prefer.

- a) Back up your SCM data.
- b) 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). |

```
cp -Rpfv /svnroot /cvsroot /tmp/backup_dir
```

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

5. Move the collabnet repository of the older version of TeamForge.

```
mv /etc/yum.repos.d/collabnet-7.0.0.x.repo /etc/yum.repos.d/
collabnet-7.0.0.x.repo.cn_backup
```

 **Note:** Replace "x" with the appropriate patch release number if applicable.

6. Run the following commands to upgrade Red Hat/CentOS to the latest version.

- a) Remove the neon-devel package if you are upgrading from Teamforge 6.2.

```
yum erase neon-devel -y
yum erase subversion-devel -y
```

- b) Upgrade the operating system packages.

```
yum upgrade
```

7. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

- a) Verify if SELinux is running in enforcing mode.


```
getenforce
```

- b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.  
c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```

See [Set up SELINUX](#) on page 271 to have TeamForge to run in SELinux mode after completing the installation or upgrade.

8. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)  
9. Uninstall the TeamForge CLI add-on (if it is already installed).

 **Note:** Skip this step if you are upgrading from TeamForge 6.2

```
cd /opt/collabnet/teamforge/add-ons/teamforge_cli_server
./install --uninstall
yum erase teamforge_cli_server
```

10. Install the following application packages.

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

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


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

```
yum install teamforge-codesearch
```

- c) If the SCM services are running on my.app.host, install the Source Code component of the TeamForge application.

```
yum install teamforge-scm
```

11. Update the `site-options.conf` file.

 **Important:** Back up your `site-options.conf` file before making any changes.

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

- a) Update the host name and domain name, if required.

```
HOST_localhost=app etl indexer
```

```
HOST_my.db.host=database datamart
```

```
HOST_my.scm.host=subversion cvs
```

```
DOMAIN_localhost=my.app.domain.com
```

- b) Add "codesearch" to `Host_localhost` if you are installing Black Duck Code Sight.

```
HOST_localhost=app etl indexer codesearch
```

- c) Configure the `JAVA_HOME` token for TeamForge.

```
JAVA_HOME=/usr/java/jdk1.7.0_40
```



- d) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

```

• SSL=on
• SSL_CERT_FILE=
• SSL_KEY_FILE=
• SSL_CA_CERT_FILE=
• SSL_CHAIN_FILE=

```

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- e) 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

```

- f) If the token `REQUIRE_PASSWORD_SECURITY` is enabled, then set a value for the token, `PASSWORD_CONTROL_EFFECTIVE_DATE`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.


- g) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any Alphanumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the [OBFUSCATION\\_PREFIX](#) on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`).

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' "`` in the `site-options.conf` file.

- h) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

```

USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer

```


- 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.


 **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 username add this
token:
BDCS_ADMIN_USERNAME=<sysadmin>
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) If you have LDAP set up for external authentication, you must set the `"REQUIRE\_USER\_PASSWORD\_CHANGE"` site options token to false.
  - l) If you are installing TeamForge through disconnected media, set the token `HELP_AVAILABILITY=local`.
  - m) Save the `site-options.conf` file.
12. Download the corresponding version of the Oracle client from <http://www.oracle.com/technetwork/database/features/instant-client/index-097480.html>

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

13. Run the following command to remove the pagespeed cache.

```
cd /opt/collabnet/teamforge/cache
rm -rf pagespeed
```

14. Back up the old TeamForge runtime directory.


```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

15. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

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. 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
ListenBackLog     2048
</IfModule>
MaxKeepAliveRequests 10000
```

c) Restart Apache.

```
/etc/init.d/httpd restart
```


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 specified by the `REPORTS_DATABASE_READ_ONLY_USER` token.

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

20. Run the following script to upgrade the *index to Lucene 4.x format*.

 **Note:** You must back up the existing search index directory before running this script. Refer to the `SEARCH_INDEX_LOCATION` token in the `runtime-options.conf` file to know the search index directory location.

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

21. Run the following script to upgrade the Subversion working copies.

```
/opt/collabnet/teamforge/runtime/scripts/svn-upgrade-working-copies.sh
```

22. 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.

23. If you are upgrading from TeamForge 7.0, run the `post-install.py` script.


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

24. If you are upgrading from TeamForge 6.2 or earlier versions and the CLI reports are already configured, follow these steps:

a) Back up the CLI reports.

```
mkdir /root/backup
cd /root/backup
backup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> find . -name .svn | xargs rm -rf
```

- b) Remove the `cli/reports` folder from the branding repository.

 **Note:** Do a SVN delete to remove this folder from the repository.

```
mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> svn delete reports
cli> svn commit -m "To delete the old CLI reports folder"
```

- c) Run the `post-install.py` script.

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

- d) Commit the backup 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```
mkdir /root/restore
cd /root/restore
restore> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
mkdir custom-reports
cp -avx /root/backup/pkg custom-reports
svn add custom-reports
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>
```

When prompted, enter the user name (<admin\_username>) and password (<admin\_password>)


```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```

25. 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.

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

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

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

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL path of the SCM integration server, for example, "`http://myint.box.net/svn/repos`", where `myint.box` is the server with the SCM integration server.

In addition, if you add a new integration server at some point later, you must run this `svn_cache.sh` script, (after creating the new integration server), on the TeamForge application server.

27. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

**28. Run the following initial load jobs (ETL).**

- 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 325.


**Do this on the SCM server (my.scm.host)**

If your TeamForge setup has the Source Control running on its own server, you'll have to upgrade that server as well.

**29. Stop TeamForge.**

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

**30. Back up your SCM data.**

 **Tip:** /tmp in the following step is just an example. You can use any directory or partition that you prefer.

- a) Make an archive with the following data directories.

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

```
cp -Rpfv /svnroot /cvsroot /tmp/scmbackup_dir
```

- b) Back up your SSH keys, if any.

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

**31. Move the collabnet repository of the older version of TeamForge.**

```
mv /etc/yum.repos.d/collabnet-7.0.0.x.repo /etc/yum.repos.d/
collabnet-7.0.0.x.repo.cn_backup
```

 **Note:** Replace "x" with the appropriate patch release number if applicable.

**32. Run the following commands to upgrade Red Hat/CentOS to the latest version.**

- a) Remove the neon-devel package if you are upgrading from Teamforge 6.2.

```
yum erase neon-devel -y
yum erase subversion-devel -y
```

- b) Upgrade the operating system packages.

```
yum upgrade
```


**33. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)**

**34. If Git is enabled and the PostgreSQL database is running locally, follow these steps.**

- a) Stop the PostgreSQL service.

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

- b) Uninstall the PostgreSQL RPMs.

 **Note:** When prompted, you must select the option to remove all the dependent packages.

```
yum erase postgresql90-libs postgresql90-docs postgresql90-server
postgresql90
```

35. Install the following application package.

a) Install the source code component.

```
yum install teamforge-scm
```

36. Copy the master `site-options.conf` file from `my.app.host` and modify the host token settings in the `site-options.conf` file.

```
HOST_localhost=subversion cvs
```

```
DOMAIN_localhost=my.scm.domain.com
```

```
HOST_my.app.host=app etl indexer codesearch
```

```
HOST_my.db.host=database datamart
```

Save the `site-options.conf` file.

37. Back up the old TeamForge runtime directory.


```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

38. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

39. 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.

40. Start the Tomcat service.

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

41. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.1.


a) Log into your site as the administrator.

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

See [Customize anything on your site](#).


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

See [Create a site-wide broadcast](#).

 **Important:** Do not delete the `teamforge-installer/7.1.0.0` directory. You will need it for future maintenance and upgrades.

## Upgrade to TeamForge 7.1 on SuSE

You can upgrade to TeamForge 7.1 from TeamForge 7.0. You can upgrade on the same server where your current TeamForge site is running, or you can take this opportunity to move your site to a new server. 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:**

- If you are upgrading from TeamForge 6.1.1 or later versions to TeamForge 7.1, you can run the `indexupgrade.py` script, which converts the Lucene 3.x indices to Lucene 4.4 format. For more information about the `indexupgrade.py` script, see [Upgrade TeamForge 7.1 search index to Lucene 4.x format](#) on page 318. However, if you choose to re-index data, it takes a lot of time and the search service would have to be down till then. Instead, if you choose to upgrade your existing indices, you can convert your site's search indices to Lucene 4.x format quickly using the `indexupgrade.py` script with less downtime of the search service.
- If you are upgrading from TeamForge 6.1 or earlier versions to TeamForge 7.1, you must re-index your site completely. This could be time consuming and depends on the size of data.

### Upgrade to TeamForge 7.1 - All Services on the same server

In this procedure, we'll assume that you are upgrading on the same server where your existing TeamForge 7.0 site is running. It is possible to simultaneously upgrade and move your site to a new hardware. However, since we are working with a dedicated installation, the priority here is to keep things as simple and quick as possible.

In this option, the following services run on the application server (we call this `my.app.host`).

- TeamForge Application Server
- Black Duck Code Sight Server
- Database Server (Operational DB and Reports DB)
- ETL Server
- GIT Integration Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer).


**Log in to the server as root.**

**Do the following on the application server - `my.app.host`**

1. Stop the Apache server and the TeamForge application server.

```
/etc/init.d/apache2 stop
/etc/init.d/collabnet stop
```

2. Back up your site database.

 **Tip:** `/tmp/backup_dir` is just an example. You can use any directory or partition you prefer to store your backup files.

- a) Create a `/backups` directory in `/var/lib/pgsql/` and change ownership to postgres.

```
cd /var/lib/pgsql/
mkdir backups
chown -R postgres:postgres backups
```

- b) Make a dump file of your site database. You have to do a PostgreSQL dump because we are upgrading the PostgreSQL application as part of this upgrade.

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

If your reporting database is running on a separate port, backup your reporting database as well.

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


Copy the database backup to the backup directory.

```
mkdir /tmp/backup_dir
cp /var/lib/pgsql/backups/teamforge_data_backup.dmp /tmp/backup_dir/
```

If your reporting database is running on a separate port, copy your reporting database dump as well.

```
cp /var/lib/pgsql/backups/teamforge_reporting_data_backup.dmp /tmp/
backup_dir/
```

### 3. Back up the file system data.

 **Tip:** /tmp/backup\_dir is just an example. You can use any directory or partition you prefer to store your backup files.

a) 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 -Rpfv /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 -Rpfv /gitroot /tmp/backup_dir
cp -Rpfv /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
backup_dir/gerrit
```

b) Back up your SSH keys, if any.

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

### 4. Move the collabnet repository of the older version of TeamForge.

```
mv /etc/zypp/repos.d/collabnet-7.0.0.x.repo /etc/zypp/repos.d/
collabnet-7.0.0.x.repo.cn_backup
```


 **Note:** Replace "x" with the appropriate patch release number if applicable.

### 5. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

### 6. If the PostgreSQL database is running locally, stop the PostgreSQL service.


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

### 7. Uninstall the TeamForge CLI add-on (if it is already installed).

 **Note:** Skip this step if you are upgrading from TeamForge 6.2

```
cd /opt/collabnet/teamforge/add-ons/teamforge_cli_server
./install --uninstall
zypper remove teamforge_cli_server
```

### 8. Uninstall the PostgreSQL RPMs.

 **Note:** When prompted, you must select the option to remove all the dependent packages.

```
zypper remove postgresql-libs postgresql-docs postgresql-server postgresql
mv /var/lib/pgsql /var/lib/pgsql_9.0
```

### 9. 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 the Black Duck Code Sight packages run the following command:

```
zypper install teamforge-codesearch
```

**10.** In the `site-options.conf` file, make sure you do the following.

-  **Note:** Back up your `site-options.conf` file before making any changes.

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

- a) Update the host name and domain name, if required.

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

```
DOMAIN_localhost=my.app.domain.com
```

Configure the following settings if you are installing Git.

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

Configure the following settings if you are installing Black Duck Code Sight.

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

- b) Configure the `JAVA_HOME` token for TeamForge.

```
JAVA_HOME=/usr/java/jdk1.7.0_40
```

- c) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

- `SSL=on`
- `SSL_CERT_FILE=`
- `SSL_KEY_FILE=`
- `SSL_CA_CERT_FILE=`
- `SSL_CHAIN_FILE=`

-  **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- 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`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- f) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to `$auto$`, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- `DATABASE_PASSWORD`
- `DATABASE_READ_ONLY_PASSWORD`
- `REPORTS_DATABASE_PASSWORD`
- `REPORTS_DATABASE_READ_ONLY_PASSWORD`
- `ETL_SOAP_SHARED_SECRET`
- `JAMES_ADMIN_PASSWORD`
- `BDCS_ADMIN_PASSWORD`
- `MIRROR_DATABASE_PASSWORD` (applicable only if you are mirroring your database)


g) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`:

 **Important:** The password-related tokens cannot contain the following characters: `$(<>/\ ' " `` in the `site-options.conf` file.

- h) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and Postgresql automatically.
- i) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

- 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
```

-  **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 username add this
token:
BDCS_ADMIN_USERNAME=<sysadmin>
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) To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
- m) If you have LDAP set up for external authentication, you must set the **"REQUIRE\_USER\_PASSWORD\_CHANGE"** site options token to false.
- n) If you are installing TeamForge through disconnected media, set the token **HELP\_AVAILABILITY=local**.
- o) Save the `site-options.conf` file.

**11.** Run the following command to remove the pagespeed cache.

```
cd /opt/collabnet/teamforge/cache
rm -rf pagespeed
```

**12.** Back up the old TeamForge runtime directory.

```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```


**13.** Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

**14.** Restore your site data.

- 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
/usr/bin/psql -p <reports_database_port> < /tmp/backup_dir/
teamforge_reporting_data_backup.dmp
exit
```

**15.** Recreate the runtime environment to set the database credentials.


```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

**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 7.1.


 **Tip:** Before you kick off the data migration, use the `/etc/init.d/collabnet status` command to make sure the Jboss, Tomcat and Tomcats 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 7.1 site.

**18.** Run the following script to upgrade the [index to Lucene 4.x format](#).

 **Note:** You must back up the existing search index directory before running this script. Refer to the `SEARCH_INDEX_LOCATION` token in the `runtime-options.conf` file to know the search index directory location.

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

**19.** Run the following script to upgrade the Subversion working copies.

```
/opt/collabnet/teamforge/runtime/scripts/svn-upgrade-working-copies.sh
```

**20.** 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.

**21.** If you are upgrading from TeamForge 7.0, run the `post-install.py` script.


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

**22.** If you are upgrading from TeamForge 6.2 or earlier versions and the CLI reports are already configured, follow these steps:

## a) Back up the CLI reports.

```
mkdir /root/backup
cd /root/backup
backup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> find . -name .svn | xargs rm -rf
```

b) Remove the `cli/reports` folder from the branding repository.

 **Note:** Do a SVN delete to remove this folder from the repository.

```
mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> svn delete reports
```

```
cli> svn commit -m "To delete the old CLI reports folder"
```

- c) Run the `post-install.py` script.

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

- d) Commit the backup 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```
mkdir /root/restore
cd /root/restore
restore> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
mkdir custom-reports
cp -avx /root/backup/pkg custom-reports
svn add custom-reports
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>
```

When prompted, enter the user name (<admin\_username>) and password (<admin\_password>)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```

23. If you have installed Git, integrate gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

- 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
```


24. Install the Black Duck Code Sight license on the server where Black Duck Code Sight is installed. For more information see [these instructions](#).

25. 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.

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

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

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

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL path of the SCM integration server, for example, "http://myint.box.net/svn/repos", where myint.box is the server with the SCM integration server.

In addition, if you add a new integration server at some point later, you must run this `svn_cache.sh` script, (after creating the new integration server), on the TeamForge application server.

27. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

28. Run the following initial load jobs (ETL).

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 325.

29. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.1.


a) Log into your site as the administrator.

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

See [Customize anything on your site](#).

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


See [Create a site-wide broadcast](#).

 **Important:** Do not delete the `teamforge-installer/7.1.0.0` directory. You will need it for future maintenance and upgrades.

30. Remove the backup files after the TeamForge site is up and running as expected.

a) Remove the repository and the file system backup from the `/tmp/backup_dir` directory.

b) Remove the PostgreSQL 9.0 database dump and the file system from the `/var/lib/pgsql/9.0/backups` and `/var/lib/pgsql/9.0/data` directories respectively.

 **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](#).

To upgrade Review Board [see these instructions](#).

### Upgrade to TeamForge 7.1 - Database and Datamart on a separate server

In this procedure, we'll assume that you are upgrading on the same server where your existing TeamForge 7.0 site is running. It is possible to simultaneously upgrade and move your site to a new hardware. However, since we are working with a dedicated installation, the priority here is to keep things as simple and quick as possible.


In this option, the following services run on the TeamForge Application Server (We call this my.app.host).

- TeamForge Application Server

- Black Duck Code Sight Server
- ETL Server
- GIT Integration Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer)

The following service runs on the Database Server (Operational DB and Reports DB). (We call this my.db.host)

- Database Server (Operational DB and Reports DB)

 **Note:** For the ETL service to run as expected in a distributed TeamForge installation, all servers must have the same time zone.


**Log in to the server as root.**

**Do the following on the application server - my.app.host**

1. Stop the Apache server and the TeamForge application server.

```
/etc/init.d/apache2 stop
/etc/init.d/collabnet stop
```

2. Back up the file system data.

 **Tip:** /tmp/backup\_dir is just an example. You can use any directory or partition you prefer to store your backup files.

- a) 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

```
mkdir -p /tmp/backup_dir
cp -Rpfv /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 -Rpfv /gitroot /tmp/backup_dir
cp -Rpfv /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
backup_dir/gerrit
```

- b) Back up your SSH keys, if any.
  - c) Back up your SSL certificates and keys, if any.
3. If Git integration is enabled, create a dump file of your gerrit's database, for PostgreSQL upgrade.
    - a) Create a /backups directory in /var/lib/pgsql/ and change ownership to postgres.

```
cd /var/lib/pgsql/
mkdir backups
chown -R postgres:postgres backups
```

- b) Create a dump file of your gerrit's database.

```
su - postgres
```


```

/usr/bin/pg_dumpall > /var/lib/pgsql/backups/
teamforge_gerrit_data_backup.dmp
exit

```

#### Do this on the database server - my.db.host

#### 4. Back up your site database.

 **Tip:** /tmp/backup\_dir is just an example. You can use any directory or partition you prefer to store your backup files.

- a) Make a dump file of your site database. You must do a PostgreSQL dump because we are upgrading the PostgreSQL application as part of this upgrade.

```

su - postgres
/usr/bin/pg_dumpall > /var/lib/pgsql/backups/teamforge_data_backup.dmp
exit

```

If your reporting database is running on a separate port, back up your reporting database as well.

```

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

```

Copy the database backup to the backup directory.

```

mkdir /tmp/dbbackup_dir
cp /var/lib/pgsql/backups/teamforge_data_backup.dmp /tmp/dbbackup_dir/

```

If your reporting database is running on a separate port, copy your reporting database dump as well.

```

cp /var/lib/pgsql/backups/teamforge_reporting_data_backup.dmp /tmp/
dbbackup_dir/

```

#### 5. Stop the PostgreSQL service.

```

/etc/init.d/postgresql stop

```

#### Do this on the application server - my.app.host

#### 6. Move the collabnet repository of the older version of TeamForge.

```


mv /etc/zypp/repos.d/collabnet-7.0.0.x.repo /etc/zypp/repos.d/
collabnet-7.0.0.x.repo.cn_backup

```

 **Note:** Replace "x" with the appropriate patch release number if applicable.

#### 7. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

#### 8. Uninstall the TeamForge CLI add-on (if it is already installed).


 **Note:** Skip this step if you are upgrading from TeamForge 6.2

```

cd /opt/collabnet/teamforge/add-ons/teamforge_cli_server
./install --uninstall
zypper remove teamforge_cli_server

```

#### 9. If Git integration is enabled, uninstall the PostgreSQL database used by gerrit as PostgreSQL may be upgraded later during the installation.

 **Note:** When prompted, you must select the option to remove all the dependent packages.

```

zypper remove postgresql-libs postgresql-docs postgresql-server postgresql
mv /var/lib/pgsql /var/lib/pgsql_9.0

```

#### 10. Install the following application packages.



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

```
zypper install teamforge-app teamforge-scm teamforge-etl
```


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

```
zypper install teamforge-git
```

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

```
zypper install teamforge-codesearch
```

**11.** In the `site-options.conf` file, make sure you do the following.

 **Note:** Back up your `site-options.conf` file before making any changes.

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

a) Update the host name and domain name, if required.

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

```
DOMAIN_localhost=my.app.domain.com
```

```
HOST_my.db.host=database datamart
```

Configure the following settings if you are installing Git.

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

Configure the following settings if you are installing Black Duck Code Sight.

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

b) Configure the `JAVA_HOME` token for TeamForge.

```
JAVA_HOME=/usr/java/jdk1.7.0_40
```

c) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

- `SSL=on`
- `SSL_CERT_FILE=`
- `SSL_KEY_FILE=`
- `SSL_CA_CERT_FILE=`
- `SSL_CHAIN_FILE=`

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

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`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- f) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to `$auto$`, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- DATABASE\_PASSWORD
- DATABASE\_READ\_ONLY\_PASSWORD
- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD
- BDCS\_ADMIN\_PASSWORD
- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


g) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`).

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- h) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and PostgreSQL automatically.
- i) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

- 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
```

-  **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 username add this
token:
BDCS_ADMIN_USERNAME=<sysadmin>
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) To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
- m) If you have LDAP set up for external authentication, you must set the “[REQUIRE\\_USER\\_PASSWORD\\_CHANGE](#)” site options token to false.
- n) If you are installing TeamForge through disconnected media, set the token *HELP\_AVAILABILITY=local*.
- o) Save the *site-options.conf* file.

**12.** Run the following command to remove the pagespeed cache.

```
cd /opt/collabnet/teamforge/cache
rm -rf pagespeed
```

**13.** Back up the old TeamForge runtime directory.

```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

**14.** Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

**15.** Restore the gerrit's PostgreSQL data.


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

**16.** Recreate the runtime environment to set the database credentials.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

**17.** 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 database server - my.db.host**

18. Move the collabnet repository of the older version of TeamForge.

```
mv /etc/zypp/repos.d/collabnet-7.0.0.x.repo /etc/zypp/repos.d/
collabnet-7.0.0.x.repo.cn_backup
```


 **Note:** Replace "x" with the appropriate patch release number if applicable.

19. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

20. If the PostgreSQL database is running locally, stop the PostgreSQL service.

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

21. Uninstall the PostgreSQL RPMs.


 **Note:** When prompted, you must select the option to remove all the dependent packages.

```
zypper remove postgresql-libs postgresql-docs postgresql-server postgresql
mv /var/lib/pgsql /var/lib/pgsql_9.0
```

22. Install the TeamForge database packages.

```
zypper install teamforge-database
```

23. Copy the `site-options.conf` file from **my.app.host** and modify the token settings.

 **Note:** If you choose to use the old `site-options.conf` file, don't forget to copy the `AUTO_DATA` token from the application server.

```
HOST_localhost=database datamart
```

```
DOMAIN_localhost=my.db.domain.com
```

```
HOST_my.app.host=app etl indexer subversion cvs
```

24. Back up the old TeamForge runtime directory.

```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```


25. Run the installer.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

26. Restore your site data.

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
/usr/bin/psql -p <reports_database_port> < /tmp/backup_dir/
teamforge_reporting_data_backup.dmp
exit
```

27. Recreate the runtime environment to set the database credentials.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

**Do this on the application server - my.app.host**

## 28. Convert your site data to work with TeamForge 7.1.


-  **Tip:** Before you kick off the data migration, use the `/etc/init.d/collabnet status` command to make sure the Jboss, Tomcat and Tomcats 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 7.1 site.

## 29. Run the following script to upgrade the *index to Lucene 4.x format*.

-  **Note:** You must back up the existing search index directory before running this script. Refer to the `SEARCH_INDEX_LOCATION` token in the `runtime-options.conf` file to know the search index directory location.

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

## 30. Run the following script to upgrade the Subversion working copies.

```
/opt/collabnet/teamforge/runtime/scripts/svn-upgrade-working-copies.sh
```

## 31. 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.

## 32. If you are upgrading from TeamForge 7.0, run the `post-install.py` script.


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

## 33. If you are upgrading from TeamForge 6.2 or earlier versions and the CLI reports are already configured, follow these steps:

### a) Back up the CLI reports.

```
mkdir /root/backup
cd /root/backup
backup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> find . -name .svn | xargs rm -rf
```

### b) Remove the `cli/reports` folder from the branding repository.

-  **Note:** Do a SVN delete to remove this folder from the repository.

```
mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> svn delete reports
cli> svn commit -m "To delete the old CLI reports folder"
```

### c) Run the `post-install.py` script.

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

### d) Commit the backup 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```

mkdir /root/restore
cd /root/restore
restore> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
mkdir custom-reports
cp -avx /root/backup/pkg custom-reports
svn add custom-reports
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>

```

When prompted, enter the user name (<admin\_username>) and password (<admin\_password>)

```

ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q

```

34. If you have installed Git, integrate Gerrit by running the `post-install.py` script.

```

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

```

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

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. 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.

```

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

```

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

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

```

cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>

```

Provide a repository base URL path of the SCM integration server, for example, "`http://myint.box.net/svn/repos`", where `myint.box` is the server with the SCM integration server.

In addition, if you add a new integration server at some point later, you must run this `svn_cache.sh` script, (after creating the new integration server), on the TeamForge application server.

37. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -v restart jboss
```

38. Run the following initial load jobs (ETL).

- 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 325.

39. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.1.


- a) Log into your site as the administrator.

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

See [Customize anything on your site](#).

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


See [Create a site-wide broadcast](#).

 **Important:** Do not delete the `teamforge-installer/7.1.0.0` directory. You will need it for future maintenance and upgrades.

40. Remove the backup files after the TeamForge site is up and running as expected.

- a) Remove the repository and the file system backup from the `/tmp/backup_dir` directory.

- b) Remove the PostgreSQL 9.0 database dump and the file system from the `/var/lib/pgsql/9.0/backups` and `/var/lib/pgsql/9.0/data` directories respectively.

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

To install TeamForge Orchestrate, see [TeamForge Orchestrate installation](#).

To upgrade Review Board [see these instructions](#).

### Upgrade to TeamForge 7.1 - Reporting services on a separate server

In this procedure, we'll assume that you are upgrading on the same server where your existing TeamForge 7.0 site is running. It is possible to simultaneously upgrade and move your site to a new hardware. However, since we are working with a dedicated installation, the priority here is to keep things as simple and quick as possible.


In this option, the following services run on the application server (we call this `my.app.host`).

- TeamForge Application Server
- Black Duck Code Sight Server
- Database Server (Operational DB)
- GIT Integration Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer)

The following service runs on the database server. (We call this `my.reports.host`)

- Database server (reports db)

- ETL Server

 **Note:** For the ETL service to run as expected in a distributed TeamForge installation, all servers must have the same time zone.


### Log in to the server as root.

#### Do the following on the application server - my.app.host

1. Stop the Apache server and the TeamForge application server.

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

2. Back up your site's operational database.

 **Tip:** /tmp/backup\_dir is just an example. You can use any directory or partition you prefer to store your backup files.

- a) Create a /backups directory in /var/lib/pgsql/ and change ownership to postgres.

```
cd /var/lib/pgsql/
mkdir backups
chown -R postgres:postgres backups
```


- b) Make a dump file of your site operational database. You have to do a PostgreSQL dump because we are upgrading the PostgreSQL application as part of this upgrade.

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

Copy the database backup to the backup directory.

```
mkdir -p /tmp/backup_dir
cp /var/lib/pgsql/backups/teamforge_data_backup.dmp /tmp/backup_dir/
```

3. Back up the file system data.

 **Tip:** /tmp/backup\_dir is just an example. You can use any directory or partition you prefer to store your backup files.

- a) 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 -Rpfv /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 -Rpfv /gitroot /tmp/backup_dir
cp -Rpfv /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
backup_dir/gerrit
```


- b) Back up your SSH keys, if any.



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

#### Do this on the reporting server - my.reports.host

4. Back up your reporting database.

 **Tip:** /tmp/backup\_dir is just an example. You can use any directory or partition you prefer to store your backup files.

```
su - postgres
/usr/bin/pg_dumpall > /var/lib/pgsql/backups/
teamforge_reporting_data_backup.dmp
exit
```

Copy the database backup to the backup directory.

```
mkdir /tmp/reportsbackup_dir
cp /var/lib/pgsql/backups/teamforge_reporting_data_backup.dmp /tmp/
reportsbackup_dir/
```

#### Do this on the application server - my.app.host

5. Move the collabnet repository of the older version of TeamForge.

```
mv /etc/zypp/repos.d/collabnet-7.0.0.x.repo /etc/zypp/repos.d/
collabnet-7.0.0.x.repo.cn_backup
```


 **Note:** Replace "x" with the appropriate patch release number if applicable.

6. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

7. If the PostgreSQL database is running locally, stop the PostgreSQL service.


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

8. Uninstall the TeamForge CLI add-on (if it is already installed).

 **Note:** Skip this step if you are upgrading from TeamForge 6.2

```
cd /opt/collabnet/teamforge/add-ons/teamforge_cli_server
./install --uninstall
zypper remove teamforge_cli_server
```

9. Uninstall the PostgreSQL RPMs.

 **Note:** When prompted, you must select the option to remove all the dependent packages.

```
zypper remove postgresql-libs postgresql-docs postgresql-server postgresql
mv /var/lib/pgsql /var/lib/pgsql_9.0
```

10. Install the following application packages.

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

```
zypper install teamforge-app teamforge-database teamforge-scm
```


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

```
zypper install teamforge-git
```

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

```
zypper install teamforge-codesearch
```

11. In the site-options.conf file, make sure you do the following.

 **Note:** Back up your site-options.conf file before making any changes.

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

- a) Update the host name and domain name, if required.

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

```
HOST_my.db.host=datamart etl
```

```
DOMAIN_localhost=my.app.domain.com
```

Configure the following settings if you are installing Git.

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

Configure the following settings if you are installing Black Duck Code Sight.

```
HOST_my.host.name=app database indexer subversion cvs codesearch
```

- b) Configure the `JAVA_HOME` token for TeamForge.

```
JAVA_HOME=/usr/java/jdk1.7.0_40
```

- c) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

- `SSL=on`
- `SSL_CERT_FILE=`
- `SSL_KEY_FILE=`
- `SSL_CA_CERT_FILE=`
- `SSL_CHAIN_FILE=`

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- 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`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- f) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to `$auto$`, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- DATABASE\_PASSWORD
- DATABASE\_READ\_ONLY\_PASSWORD
- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD
- BDCS\_ADMIN\_PASSWORD
- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


g) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF};`.

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- h) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and PostgreSQL automatically.
- i) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

- 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
```

 **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 username add this
token:
BDCS_ADMIN_USERNAME=<sysadmin>
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) To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
- m) If you have LDAP set up for external authentication, you must set the **"REQUIRE\_USER\_PASSWORD\_CHANGE"** site options token to false.
- n) If you are installing TeamForge through disconnected media, set the token **HELP\_AVAILABILITY=local**.
- o) Save the `site-options.conf` file.

12. Run the following command to remove the pagespeed cache.

```
cd /opt/collabnet/teamforge/cache
rm -rf pagespeed
```

13. Back up the old TeamForge runtime directory.

```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

14. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

15. Restore your site data.

- a) Reload the PostgreSQL data.


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

16. Recreate the runtime environment to set the database credentials.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

17. 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 reporting server - my.reports.host**

18. Move the collabnet repository of the older version of TeamForge.

```
mv /etc/zypp/repos.d/collabnet-7.0.0.x.repo /etc/zypp/repos.d/
collabnet-7.0.0.x.repo.cn_backup
```


 **Note:** Replace "x" with the appropriate patch release number if applicable.

19. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

20. If the PostgreSQL database is running locally, stop the PostgreSQL service.

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

**21. Uninstall the PostgreSQL RPMs.**


 **Note:** When prompted, you must select the option to remove all the dependent packages.

```
zypper remove postgresql-libs postgresql-docs postgresql-server postgresql
mv /var/lib/pgsql /var/lib/pgsql_9.0
```

**22. Install the TeamForge database packages.**

```
zypper remove teamforge-database teamforge-etl
```

**23. Copy the `site-options.conf` file from `my.app.host` and modify the token settings.**

 **Note:** If you choose to use the old `site-options.conf` file, don't forget to copy the `AUTO_DATA` token from the application server.

```
HOST_localhost=datamart etl
```

```
DOMAIN_localhost=my.reports.domain.com
```

```
HOST_my.app.host=app database indexer subversion cvs
```

**24. Back up the old TeamForge runtime directory.**

```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

**25. Run the installer.**

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

**26. Restore your reports data.**


a) Reload the PostgreSQL data.

```
su - postgres
/usr/bin/psql < /tmp/reportsbackup_dir/
teamforge_reporting_data_backup.dmp
exit
```

**27. Recreate the runtime environment to set the database credentials.**

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

**Do this on the application server - `my.app.host`****28. Convert your site data to work with TeamForge 7.1.**


 **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 7.1 site.

**29. Run the following script to upgrade the [index to Lucene 4.x format](#).**

 **Note:** You must back up the existing search index directory before running this script. Refer to the `SEARCH_INDEX_LOCATION` token in the `runtime-options.conf` file to know the search index directory location.

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

**30. Run the following script to upgrade the Subversion working copies.**

```
/opt/collabnet/teamforge/runtime/scripts/svn-upgrade-working-copies.sh
```

### 31. 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.

### Do this on the reporting server - my.reports.host

#### 32. Start the ETL service.

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

### Do the following on the application server - my.app.host

#### 33. If you are upgrading from TeamForge 7.0, run the `post-install.py` script.

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

#### 34. If you are upgrading from TeamForge 6.2 or earlier versions and the CLI reports are already configured, follow these steps:

##### a) Back up the CLI reports.

```
mkdir /root/backup
cd /root/backup
backup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> find . -name .svn | xargs rm -rf
```

##### b) Remove the `cli/reports` folder from the branding repository.

#### **Note:** Do a SVN delete to remove this folder from the repository.

```
mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> svn delete reports
cli> svn commit -m "To delete the old CLI reports folder"
```

##### c) Run the `post-install.py` script.

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

##### d) Commit the backup 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```
mkdir /root/restore
cd /root/restore
restore> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
mkdir custom-reports
cp -avx /root/backup/pkg custom-reports
svn add custom-reports
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>

```

When prompted, enter the user name (<admin\_username>) and password (<admin\_password>)

```

ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q

```

35. If you have installed Git, integrate Gerrit by running the `post-install.py` script.

```

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

```

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

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

```

36. 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.

```

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

```

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

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

```

cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>

```

Provide a repository base URL path of the SCM integration server, for example, "`http://myint.box.net/svn/repos`", where `myint.box` is the server with the SCM integration server.

In addition, if you add a new integration server at some point later, you must run this `svn_cache.sh` script, (after creating the new integration server), on the TeamForge application server.

38. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```

cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss

```

**Do this on the reporting server - my.reports.host**

39. Run the following initial load jobs (ETL).

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 325.


### Do the following on the application server - my.app.host

40. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.1.

- a) Log into your site as the administrator.
- b) If your site has custom branding, verify that your branding changes still work as intended.


See [Customize anything on your site](#).
- c) Let your site's users know they've been upgraded.

See [Create a site-wide broadcast](#).

 **Important:** Do not delete the `teamforge-installer/7.1.0.0` directory. You will need it for future maintenance and upgrades.

41. Remove the backup files after the TeamForge site is up and running as expected.

- a) Remove the repository and the file system backup from the `/tmp/backup_dir` directory.
- b) Remove the PostgreSQL 9.0 database dump and the file system from the `/var/lib/pgsql/9.0/backups` and `/var/lib/pgsql/9.0/data` directories respectively.

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

To install TeamForge Orchestrate, see [TeamForge Orchestrate installation](#).

To upgrade Review Board [see these instructions](#).

### Upgrade to TeamForge 7.1 - Black Duck Code Sight on a separate server


In this procedure, we'll assume that you are upgrading on the same server where your existing TeamForge site is running on 7.0. It is possible to simultaneously upgrade and move your site to a new hardware. However, since we are working with a dedicated installation, the priority here is to keep things as simple and quick as possible.

In this option, the following services run on the application server (we call this my.app.host).

- TeamForge Application Server
- ETL Server
- Database Server (Operational DB and Reports DB)
- GIT Integration Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer)

The following service runs on the Code Sight server. (We call this my.codesight.host)

- Code Sight server

 **Note:** For the ETL service to run as expected in a distributed TeamForge installation, all servers must have the same time zone.

### Log in to the server as root.


#### Do the following on the application server - my.app.host

1. Stop the Apache server and the TeamForge application server.

```
/etc/init.d/apache2 stop
/etc/init.d/collabnet stop
```



## 2. Back up your site database.

 **Tip:** /tmp/backup\_dir is just an example. You can use any directory or partition you prefer to store your backup files.

- a) Create a /backups directory in /var/lib/pgsql/ and change ownership to postgres.

```
cd /var/lib/pgsql/
mkdir backups
chown -R postgres:postgres backups
```

- b) Make a dump file of your site database. You have to do a PostgreSQL dump because we are upgrading the PostgreSQL application as part of this upgrade.

```
su - postgres
/usr/bin/pg_dumpall > /var/lib/pgsql/backups/teamforge_data_backup.dmp
```

If your reporting database is running on a separate port, backup your reporting database as well.

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


Copy the database backup to the backup directory.

```
mkdir /tmp/backup_dir
cp /var/lib/pgsql/backups/teamforge_data_backup.dmp /tmp/backup_dir/
```

If your reporting database is running on a separate port, copy your reporting database dump as well.

```
cp /var/lib/pgsql/backups/teamforge_reporting_data_backup.dmp /tmp/
backup_dir/
```

## 3. Back up the file system data.

 **Tip:** /tmp in the following step is just an example. You can use any directory or partition that you prefer.

- a) 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 -Rpfv /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 -Rpfv /gitroot /tmp/backup_dir
cp -Rpfv /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
backup_dir/gerrit
```

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

## 4. Move the collabnet repository of the older version of TeamForge.

```
mv /etc/zypp/repos.d/collabnet-7.0.0.x.repo /etc/zypp/repos.d/
collabnet-7.0.0.x.repo.cn_backup
```


 **Note:** Replace "x" with the appropriate patch release number if applicable.

5. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

6. If the PostgreSQL database is running locally, stop the PostgreSQL service.


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

7. Uninstall the TeamForge CLI add-on (if it is already installed).

 **Note:** Skip this step if you are upgrading from TeamForge 6.2

```
cd /opt/collabnet/teamforge/add-ons/teamforge_cli_server
./install --uninstall
zypper remove teamforge_cli_server
```

8. Uninstall the PostgreSQL RPMs.

 **Note:** When prompted, you must select the option to remove all the dependent packages.

```
zypper remove postgresql-libs postgresql-docs postgresql-server postgresql
mv /var/lib/pgsql /var/lib/pgsql_9.0
```

9. 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
```

10. In the `site-options.conf` file, make sure you do the following.

 **Note:** Back up your `site-options.conf` file before making any changes.

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

a) Update the host name and domain name, if required.

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

```
DOMAIN_localhost=my.app.domain.com
```

```
HOST_my.codesight.host=codesearch
```

Configure the following settings if you are installing Git.

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

b) Configure the `JAVA_HOME` token for TeamForge.

```
JAVA_HOME=/usr/java/jdk1.7.0_40
```

- c) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

- `SSL=on`
- `SSL_CERT_FILE=`
- `SSL_KEY_FILE=`
- `SSL_CA_CERT_FILE=`
- `SSL_CHAIN_FILE=`

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- 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`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- f) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to `$auto$`, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- `DATABASE_PASSWORD`
- `DATABASE_READ_ONLY_PASSWORD`
- `REPORTS_DATABASE_PASSWORD`
- `REPORTS_DATABASE_READ_ONLY_PASSWORD`
- `ETL_SOAP_SHARED_SECRET`
- `JAMES_ADMIN_PASSWORD`
- `BDCS_ADMIN_PASSWORD`
- `MIRROR_DATABASE_PASSWORD` (applicable only if you are mirroring your database)


- g) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`.

 **Important:** The password-related tokens cannot contain the following characters: \$<>/\ ' " ` in the `site-options.conf` file.

- h) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and PostgreSQL automatically.
- i) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

- 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.codesight.host>
```

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 username add this token:

```
BDCS_ADMIN_USERNAME=<sysadmin>
```

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) To enable the history protection feature of TeamForge Git integration, set the `GERRIT_FORCE_HISTORY_PROTECTION=true`. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
- m) If you have LDAP set up for external authentication, you must set the `"REQUIRE_USER_PASSWORD_CHANGE"` site options token to false.
- n) If you are installing TeamForge through disconnected media, set the token `HELP_AVAILABILITY=local`.
- o) Save the `site-options.conf` file.

11. Run the following command to remove the pagespeed cache.

```
cd /opt/collabnet/teamforge/cache
rm -rf pagespeed
```

## 12. Back up the old TeamForge runtime directory.

```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```


## 13. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

## 14. Restore your site data.

### 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
/usr/bin/psql -p <reports_database_port> < /tmp/backup_dir/
teamforge_reporting_data_backup.dmp
exit
```

## 15. Recreate the runtime environment to set the database credentials.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

## 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 the my.codesight.host

## 17. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)


## 18. Stop the Black Duck Code Sight service.

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

## 19. Install Black Duck Code Sight.

```
zypper install teamforge-codesearch
```

## 20. Copy the master site-options.conf file from my.app.host and modify the token settings.

 **Note:** If you choose to use the old site-options.conf file, don't forget to copy the *AUTO\_DATA* token from the application server.

```
HOST_localhost=codesearch
```

```
DOMAIN_localhost=my.codesight.domain.com
```

```
Host_my.app.host=app database datamart etl indexer subversion cvs
```

Save the site-options.conf file.

## 21. Back up the old TeamForge runtime directory.

```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

## 22. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

23. Run the following script to upgrade the Subversion working copies.

```
/opt/collabnet/teamforge/runtime/scripts/svn-upgrade-working-copies.sh
```


24. Start the Black Duck Code Sight service.

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

25. To install the license for Black Duck Code Sight follow [these instructions](#).

**Do this on my.app.host**

26. Convert your site data to work with TeamForge 7.1.


 **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 7.1 site.

27. Run the following script to upgrade the [index to Lucene 4.x format](#).

 **Note:** You must back up the existing search index directory before running this script. Refer to the `SEARCH_INDEX_LOCATION` token in the `runtime-options.conf` file to know the search index directory location.

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

28. Run the following script to upgrade the Subversion working copies.

```
/opt/collabnet/teamforge/runtime/scripts/svn-upgrade-working-copies.sh
```

29. 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.

30. If you are upgrading from TeamForge 7.0, run the `post-install.py` script.


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

31. If you are upgrading from TeamForge 6.2 or earlier versions and the CLI reports are already configured, follow these steps:

a) Back up the CLI reports.

```
mkdir /root/backup
cd /root/backup
backup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> find . -name .svn | xargs rm -rf
```

b) Remove the `cli/reports` folder from the branding repository.

 **Note:** Do a SVN delete to remove this folder from the repository.

```

mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> svn delete reports
cli> svn commit -m "To delete the old CLI reports folder"

```

- c) Run the `post-install.py` script.

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

- d) Commit the backup 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```

mkdir /root/restore
cd /root/restore
restore> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
mkdir custom-reports
cp -avx /root/backup/pkg custom-reports
svn add custom-reports
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>

```

When prompted, enter the user name (<admin\_username>) and password (<admin\_password>)

```

ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q

```

32. If you have installed Git, integrate gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

- 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
```

### Do this on my.codesight.host

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.

```
/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:

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

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL path of the SCM integration server, for example, "`http://myint.box.net/svn/repos`", where `myint.box` is the server with the SCM integration server.

In addition, if you add a new integration server at some point later, you must run this `svn_cache.sh` script, (after creating the new integration server), on the TeamForge application server.

### Do this on my.app.host

35. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

36. Run the following initial load jobs (ETL).

- 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 325.

37. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.1.

- a) Log into your site as the administrator.
- b) If your site has custom branding, verify that your branding changes still work as intended.  
See [Customize anything on your site](#).
- c) Let your site's users know they've been upgraded.  
See [Create a site-wide broadcast](#).

- 👉 **Important:** Do not delete the `teamforge-installer/7.1.0.0` directory. You will need it for future maintenance and upgrades.

38. Remove the backup files after the TeamForge site is up and running as expected.

- a) Remove the repository and the file system backup from the `/tmp/backup_dir` directory.
- b) Remove the PostgreSQL 9.0 database dump and the file system from the `/var/lib/pgsql/9.0/backups` and `/var/lib/pgsql/9.0/data` directories respectively.

- 👉 **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](#).

To upgrade Review Board [see these instructions](#).



## Upgrade to TeamForge 7.1 - GIT on a separate server


In this procedure, we'll assume that you are upgrading on the same server where your existing TeamForge site is running on 7.0. It is possible to simultaneously upgrade and move your site to a new hardware. However, since we are working with a dedicated installation, the priority here is to keep things as simple and quick as possible.

In this option, the following services run on the application server (we call this my.app.host).

- TeamForge Application Server
- Black Duck Code Sight Server
- ETL Server
- Database Server (Operational DB and Reports DB)
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer)

The following service runs on the GIT Integration Server. (We call this my.git.host)

- GIT Integration Server

 **Note:** For the ETL service to run as expected in a distributed TeamForge installation, all servers must have the same time zone.

**Log in to the server as root.**

**Do the following on the application server - my.app.host**

1. Stop the Apache server and the TeamForge application server.


```
/etc/init.d/apache2 stop
/etc/init.d/collabnet stop
```

2. Back up your site database.

- a) Create a /backups directory in /var/lib/pgsql/ and change ownership to postgres.

```
cd /var/lib/pgsql/
mkdir backups
chown -R postgres:postgres backups
```

- b) Make a dump file of your site database. You have to do a PostgreSQL dump because we are upgrading the PostgreSQL application as part of this upgrade.

 **Tip:** /tmp/backup\_dir is just an example. You can use any directory or partition you prefer to store your backup files.

```
su - postgres
/usr/bin/pg_dumpall > /var/lib/pgsql/backups/teamforge_data_backup.dmp
```

If your reporting database is running on a separate port, backup your reporting database as well.

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


Copy the database backup to the backup directory.

```
mkdir /tmp/backup_dir
cp /var/lib/pgsql/backups/teamforge_data_backup.dmp /tmp/backup_dir/
```

If your reporting database is running on a separate port, copy your reporting database dump as well.

```
cp /var/lib/pgsql/teamforge_reporting_data_backup.dmp /tmp/backup_dir/
```

3. Back up the file system data.

 **Tip:** /tmp/backup\_dir is just an example. You can use any directory or partition you prefer to store your backup files.

- a) 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 -Rpfv /svnroot /sf-svnroot /cvsroot /opt/collabnet/teamforge/var /
tmp/backup_dir
```

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

4. Move the collabnet repository of the older version of TeamForge.

```
mv /etc/zypp/repos.d/collabnet-7.0.0.x.repo /etc/zypp/repos.d/
collabnet-7.0.0.x.repo.cn_backup
```


 **Note:** Replace "x" with the appropriate patch release number if applicable.

5. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

6. If the PostgreSQL database is running locally, stop the PostgreSQL service.


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

7. Uninstall the TeamForge CLI add-on (if it is already installed).

 **Note:** Skip this step if you are upgrading from TeamForge 6.2

```
cd /opt/collabnet/teamforge/add-ons/teamforge_cli_server
./install --uninstall
zypper remove teamforge_cli_server
```

8. Uninstall the PostgreSQL RPMs.

 **Note:** When prompted, you must select the option to remove all the dependent packages.

```
zypper remove postgresql-libs postgresql-docs postgresql-server postgresql
mv /var/lib/pgsql /var/lib/pgsql_9.0
```

9. 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 the Black Duck Code Sight packages run the following command:

```
zypper install teamforge-codesearch
```

10. In the `site-options.conf` file, make sure you do the following.

 **Note:** Back up your `site-options.conf` file before making any changes.

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

- a) Update the host name and domain name, if required.

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

```
DOMAIN_localhost=my.app.domain.com
```

```
HOST_my.git.host=gerrit
```

Configure the following settings if you are installing Black Duck Code Sight.

```
HOST_my.localhost=app database datamart etl indexer subversion
cvs codesearch
```

- b) Configure the `JAVA_HOME` token for TeamForge.

```
JAVA_HOME=/usr/java/jdk1.7.0_40
```

- c) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

- `SSL=on`
- `SSL_CERT_FILE=`
- `SSL_KEY_FILE=`
- `SSL_CA_CERT_FILE=`
- `SSL_CHAIN_FILE=`

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- 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`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- f) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to `$auto$`, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- `DATABASE_PASSWORD`
- `DATABASE_READ_ONLY_PASSWORD`

- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD
- BDCS\_ADMIN\_PASSWORD
- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


#### g) Password Obfuscation

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`).

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

h) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and PostgreSQL automatically.

i) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

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
```

 **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 username add this token:

```
BDCS_ADMIN_USERNAME=<sysadmin>
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) To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
- m) If you have LDAP set up for external authentication, you must set the “[REQUIRE\\_USER\\_PASSWORD\\_CHANGE](#)” site options token to false.
- n) If you are installing TeamForge through disconnected media, set the token `HELP_AVAILABILITY=local`.
- o) Save the `site-options.conf` file.

**11. Run the following command to remove the pagespeed cache.**

```
cd /opt/collabnet/teamforge/cache
rm -rf pagespeed
```

**12. Back up the old TeamForge runtime directory.**

```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```


**13. Recreate the runtime environment.**

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

**14. Restore your site data.**

- 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
/usr/bin/psql -p <reports_database_port> < /tmp/backup_dir/
teamforge_reporting_data_backup.dmp
exit
```

**15. Recreate the runtime environment to set the database credentials.**


```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

**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 7.1.**


 **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 7.1 site.

18. Run the following script to upgrade the *index to Lucene 4.x format*.

-  **Note:** You must back up the existing search index directory before running this script. Refer to the `SEARCH_INDEX_LOCATION` token in the `runtime-options.conf` file to know the search index directory location.

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

19. Run the following script to upgrade the Subversion working copies.

```
/opt/collabnet/teamforge/runtime/scripts/svn-upgrade-working-copies.sh
```

20. 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.

21. If you are upgrading from TeamForge 7.0, run the `post-install.py` script.


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

22. If you are upgrading from TeamForge 6.2 or earlier versions and the CLI reports are already configured, follow these steps:

a) Back up the CLI reports.

```
mkdir /root/backup
cd /root/backup
backup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> find . -name .svn | xargs rm -rf
```

b) Remove the `cli/reports` folder from the branding repository.

-  **Note:** Do a SVN delete to remove this folder from the repository.

```
mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> svn delete reports
cli> svn commit -m "To delete the old CLI reports folder"
```

c) Run the `post-install.py` script.

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

d) Commit the backup 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```
mkdir /root/restore
cd /root/restore
restore> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
mkdir custom-reports
cp -avx /root/backup/pkg custom-reports
svn add custom-reports
```

```
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>
```

When prompted, enter the user name (<admin\_username>) and password (<admin\_password>)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```


23. Install the Black Duck Code Sight license on the server where Black Duck Code Sight is installed. For more information see [these instructions](#).

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.

```
/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:

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

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL path of the SCM integration server, for example, "`http://myint.box.net/svn/repos`", where `myint.box` is the server with the SCM integration server.

In addition, if you add a new integration server at some point later, you must run this `svn_cache.sh` script, (after creating the new integration server), on the TeamForge application 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 server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

27. Run the following initial load jobs (ETL).

- 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 325.

### Do this on the Git integration server - `my.git.host`

28. Back up your Git database.

- a) Make a dump file of your Git database. You have to do a `PostgreSQLdump` because we are upgrading the PostgreSQL application as part of this upgrade.

 **Tip:** `/tmp` is just an example. You can use any directory or partition that you prefer.


```

su-postgres
/usr/bin/pg_dumpall > /var/lib/pgsql/backups/git_data_backup.dmp
exit
mkdir -p /tmp/gitbackup_dir/gerrit
cp /var/lib/pgsql/backups/git_data_backup.dmp /tmp/gitbackup_dir/

```

### 29. Back up the Git file system data.

- a) Make an archive file with the following data directories.

 **Tip:** /tmp is just an example. You can use any directory or partition that you prefer.

| Directory | Contents                     |
|-----------|------------------------------|
| /gitroot  | Git source code repositories |

```

mkdir /tmp/backup_dir/gerrit
cp -Rpfv /gitroot /tmp/backup_dir
cp -Rpfv /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
backup_dir/gerrit

```

- b) Back up your SSH keys, if any.

### 30. Move the collabnet repository of the older version of TeamForge.

```


mv /etc/zypp/repos.d/collabnet-7.0.0.x.repo /etc/zypp/repos.d/
collabnet-7.0.0.x.repo.cn_backup

```

 **Note:** Replace "x" with the appropriate patch release number if applicable.

### 31. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

### 32. Uninstall the PostgreSQL RPMs.

 **Note:** When prompted, you must select the option to remove all the dependent packages.

```


zypper remove postgresql-libs postgresql-docs postgresql-server postgresql
mv /var/lib/pgsql /var/lib/pgsql_9.0

```

### 33. Install the Git packages.

```
zypper install teamforge-git
```

### 34. Copy the site-options.conf file from my.app.host and modify the token settings.

 **Note:** If you choose to use the old site-options.conf file, don't forget to copy the *AUTO\_DATA* token from the application server.

```
HOST_localhost=gerrit
```

```
DOMAIN_localhost=my.git.domain.com
```

```
HOST_my.app.host=app database datamart etl indexer subversion cvs
```

### 35. Back up the old TeamForge runtime directory.

```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

### 36. Run the installer.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

### 37. Restore your Git data (if Git is enabled).



- a) Reload the PostgreSQL data.

```
su - postgres
/usr/bin/psql < /tmp/gitbackup_dir/git_data_backup.dmp
exit
```

38. Run the installer.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

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

```
/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:

Login to the app server and run the following command:

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


40. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.1.

- Log into your site as the administrator.
- If your site has custom branding, verify that your branding changes still work as intended.  
See [Customize anything on your site](#).
- Let your site's users know they've been upgraded.  
See [Create a site-wide broadcast](#).

 **Important:** Do not delete the `teamforge-installer/7.1.0.0` directory. You will need it for future maintenance and upgrades.

41. Remove the backup files after the TeamForge site is up and running as expected.

- Remove the repository and the file system backup from the `/tmp/backup_dir` directory.
- Remove the PostgreSQL 9.0 database dump and the file system from the `/var/lib/pgsql/9.0/backups` and `/var/lib/pgsql/9.0/data` directories respectively.

 **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](#).

To upgrade Review Board [see these instructions](#).

### Upgrade to TeamForge 7.1 on new hardware - All Services on the same server

To upgrade to TeamForge 7.1, set up a new hardware, then bring your old site's data and convert it.

In this option, the following services run on the application server (we call this `my.app.host`).

- TeamForge Application Server
- Black Duck Code Sight Server
- Database Server (Operational DB and Reports DB)

- ETL Server
- GIT Integration Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer).


### Log in to the server as root.

### Do the following on the existing TeamForge application server - my.app.host

1. Stop the Apache server and the TeamForge application server.

```
/etc/init.d/apache2 stop
/etc/init.d/collabnet stop
```

2. Back up your site database.

 **Tip:** /tmp/backup\_dir is just an example. You can use any directory or partition you prefer to store your backup files.

- a) Create a /backups directory in /var/lib/pgsql/ and change ownership to postgres.

```
cd /var/lib/pgsql/
mkdir backups
chown -R postgres:postgres backups
```

- b) Make a dump file of your site database. You have to do a PostgreSQL dump because we are upgrading the PostgreSQL application as part of this upgrade.

```
su - postgres
/usr/bin/pg_dumpall > /var/lib/pgsql/backups/teamforge_data_backup.dmp
```

If your reporting database is running on a separate port, backup your reporting database as well.

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

Copy the database backup to the backup directory.


```
mkdir /tmp/backup_dir
cp /var/lib/pgsql/backups/teamforge_data_backup.dmp /tmp/backup_dir/
```

If your reporting database is running on a separate port, copy your reporting database dump as well.


```
cp /var/lib/pgsql/backups/teamforge_reporting_data_backup.dmp /tmp/
backup_dir/
```

3. If you have Black Duck Code Sight installed, then back up the Black Duck Code Sight data.

```
cd /opt/collabnet
tar czvf /tmp/backup_dir/blackduck.tgz blackduck
```

 **Tip:** If the Black Duck Code Sight directory size is huge, the back up task may run for longer duration. You may proceed with the following steps while the Black Duck Code Sight is being backed up.

4. Back up the file system data.

 **Tip:** /tmp/backup\_dir is just an example. You can use any directory or partition you prefer to store your backup files.

- a) 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             |

| Directory   | Contents  |
|-------------|---|
| /sf-svnroot | Subversion repository for branding data                 |
| /cvsroot    | CVS source code repositories (not present on all sites) |
| /gitroot    | GIT source code repositories                            |

```
cp -Rpfv /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 -Rpfv /gitroot /tmp/backup_dir
cp -Rpfv /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/
backup_dir/gerrit
```

Compress your backup data.


```
cd /tmp
tar czvf 70backup.tgz backup_dir
```

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

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

```
scp /opt/collabnet/teamforge-installer/7.0.0.x/conf/site-options.conf
username@newbox:/tmp
```

 **Note:** Replace "x" with the appropriate patch release number if applicable.

 **Tip:** scp is just an example. You can choose any file transfer method you prefer.

6. Copy the file system data to the new server.

```
scp /tmp/70backup.tgz username@newbox:/tmp
```

**Do the following on the new TeamForge Application Server**

7. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

8. 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 the Black Duck Code Sight packages run the following command:

```
zypper install teamforge-codesearch
```


9. Copy the `site-options.conf` file to the TeamForge installer directory.

```
cp /tmp/site-options.conf /opt/collabnet/teamforge-installer/7.1.0.0/conf
```

10. Unpack the file system data.

```
cd /tmp
tar xzvf 70backup.tgz
```

11. In the `site-options.conf` file, make sure you do the following.

 **Note:** Back up your `site-options.conf` file before making any changes.

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

- a) Update the host name and domain name, if required.

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

```
DOMAIN_localhost=my.app.domain.com
```

Configure the following settings if you are installing Git.

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

Configure the following settings if you are installing Black Duck Code Sight.

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

- b) Configure the `JAVA_HOME` token for TeamForge.

```
JAVA_HOME=/usr/java/jdk1.7.0_40
```

- c) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

- `SSL=on`
- `SSL_CERT_FILE=`
- `SSL_KEY_FILE=`
- `SSL_CA_CERT_FILE=`
- `SSL_CHAIN_FILE=`

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- 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`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- f) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to `$auto$`, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- DATABASE\_PASSWORD
- DATABASE\_READ\_ONLY\_PASSWORD
- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD
- BDCS\_ADMIN\_PASSWORD
- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


g) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF};`.

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- h) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and Postgresql automatically.
- i) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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


If you are installing Git, 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
```

- 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.app.host or my.domain.com>
```

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 username add this
token:
BDCS_ADMIN_USERNAME=<sysadmin>
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) To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
- m) If you have LDAP set up for external authentication, you must set the “[REQUIRE\\_USER\\_PASSWORD\\_CHANGE](#)” site options token to false.
- n) If you are installing TeamForge through disconnected media, set the token *HELP\_AVAILABILITY=local*.
- o) Save the `site-options.conf` file.


## 12. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

## 13. Restore your site data.

- 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
/usr/bin/psql -p <reports_database_port> < /tmp/backup_dir/
teamforge_reporting_data_backup.dmp
exit
```

- b) If you have installed Black Duck Code Sight, restore the Black Duck Code Sight data.

```
cd /opt/collabnet
mv blackduck blackduck_bak
tar -zxvf /tmp/backup_dir/blackduck.tgz
```

## 14. Reload the svnroot, sf-svnroot, cvsroot, gitroot and var directories.

```
cp -Rpfv /tmp/backup_dir/svnroot /svnroot
cp -Rpfv /tmp/backup_dir/cvsroot /cvsroot
cp -Rpfv /tmp/backup_dir/sf-svnroot /sf-svnroot
cp -Rpfv /tmp/backup_dir/var /opt/collabnet/teamforge/var
```

If Git integration is enabled, do the following:


```
cp -Rpfv /tmp/backup_dir/gitroot /
cp -Rpfv /tmp/backup_dir/gerrit/etc /opt/collabnet/gerrit
cp -Rpf /tmp/backup_dir/gerrit/.ssh /opt/collabnet/gerrit
```

## 15. Recreate the runtime environment to set the database credentials.


```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

**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 7.1.


 **Tip:** Before you kick off the data migration, use the `/etc/init.d/collabnet status` command to make sure the Jboss, Tomcat and Tomcats 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 7.1 site.

**18.** Run the following script to upgrade the [index to Lucene 4.x format](#).

 **Note:** You must back up the existing search index directory before running this script. Refer to the `SEARCH_INDEX_LOCATION` token in the `runtime-options.conf` file to know the search index directory location.


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

**19.** Run the following script to upgrade the Subversion working copies.

```
/opt/collabnet/teamforge/runtime/scripts/svn-upgrade-working-copies.sh
```

**20.** 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.

**21.** If you are upgrading from TeamForge 7.0, run the `post-install.py` script.


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

**22.** If you are upgrading from TeamForge 6.2 or earlier versions and the CLI reports are already configured, follow these steps:

## a) Back up the CLI reports.

```
mkdir /root/backup
cd /root/backup
backup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> find . -name .svn | xargs rm -rf
```

b) Remove the `cli/reports` folder from the branding repository.

 **Note:** Do a SVN delete to remove this folder from the repository.

```
mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> svn delete reports
```

```
cli> svn commit -m "To delete the old CLI reports folder"
```

- c) Run the `post-install.py` script.

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

- d) Commit the backup 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```
mkdir /root/restore
cd /root/restore
restore> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
mkdir custom-reports
cp -avx /root/backup/pkg custom-reports
svn add custom-reports
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>
```

When prompted, enter the user name (<admin\_username>) and password (<admin\_password>)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```

23. If you have installed Git, integrate gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

- 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
```

24. Install the Black Duck Code Sight license on the server where Black Duck Code Sight is installed. For more information see [these instructions](#).


25. 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.

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

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



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

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL path of the SCM integration server, for example, "http://myint.box.net/svn/repos", where myint.box is the server with the SCM integration server.

In addition, if you add a new integration server at some point later, you must run this `svn_cache.sh` script, (after creating the new integration server), on the TeamForge application server.

27. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

28. Run the following initial load jobs (ETL).

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 325.

29. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.1.


a) Log into your site as the administrator.

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

See [Customize anything on your site](#).

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


See [Create a site-wide broadcast](#).

 **Important:** Do not delete the `teamforge-installer/7.1.0.0` directory. You will need it for future maintenance and upgrades.

30. Remove the backup files after the TeamForge site is up and running as expected.

a) Remove the repository and the file system backup from the `/tmp/backup_dir` directory.

b) Remove the PostgreSQL 9.0 database dump and the file system from the `/var/lib/pgsql/9.0/backups` and `/var/lib/pgsql/9.0/data` directories respectively.

 **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](#).

To upgrade Review Board [see these instructions](#).

### Upgrade to TeamForge 7.1 - Database and SCM on separate servers

In this procedure, we'll assume that you are upgrading on the same server where your existing TeamForge 7.0 site is running. It is possible to simultaneously upgrade and move your site to a new hardware. However, since we are working with a dedicated installation, the priority here is to keep things as simple and quick as possible.

In this option, the following services run on the application server (we call this my.app.host).

- TeamForge Application Server


- Black Duck Code Sight Server
- ETL Server
- Search Server (Indexer).

The following service runs on the database server (We call this my.db.host)

- Database Server (Operational DB and Reports DB)

The following services run of the SCM server (We call this my.scm.host)

- SCM Integration Server (Subversion and CVS)
- GIT Integration Server

 **Note:** For the ETL service to run as expected in a distributed TeamForge installation, all servers must have the same time zone.


**Log in to the server as root.**

**Do the following on the application server - my.app.host**

1. Stop the Apache server and the TeamForge application server.

```
/etc/init.d/apache2 stop
/etc/init.d/collabnet stop
```

2. Back up the file system data.

 **Tip:** /tmp/backup\_dir is just an example. You can use any directory or partition you prefer to store your backup files.

- a) 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 -p /tmp/backup_dir
cp -Rpfv /sf-svnroot /opt/collabnet/teamforge/var /tmp/backup_dir
```

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

**Do this on the database server - my.db.host**

3. Back up your site database.

 **Tip:** /tmp/backup\_dir is just an example. You can use any directory or partition you prefer to store your backup files.

- a) Create a /backups directory in /var/lib/pgsql/ and change ownership to postgres.

```
cd /var/lib/pgsql/
mkdir backups
chown -R postgres:postgres backups
```

- b) Make a dump file of your site database. You have to do a PostgreSQL dump because we are upgrading the PostgreSQL application as part of this upgrade.

```
su - postgres
/usr/bin/pg_dumpall > /var/lib/pgsql/backups/teamforge_data_backup.dmp
```

If your reporting database is running on a separate port, backup your reporting database as well.

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

Copy the database backup to the backup directory.

```
mkdir /tmp/dbbackup_dir
cp /var/lib/pgsql/backups/teamforge_data_backup.dmp /tmp/dbbackup_dir/
```

If your reporting database is running on a separate port, copy your reporting database dump as well.

```
cp /var/lib/pgsql/backups/teamforge_reporting_data_backup.dmp /tmp/
dbbackup_dir/
```


#### Do this on the application server - my.app.host

4. Move the collabnet repository of the older version of TeamForge.

```
mv /etc/zypp/repos.d/collabnet-7.0.0.x.repo /etc/zypp/repos.d/
collabnet-7.0.0.x.repo.cn_backup
```

 **Note:** Replace "x" with the appropriate patch release number if applicable.

5. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)
6. Uninstall the TeamForge CLI add-on (if it is already installed).

 **Note:** Skip this step if you are upgrading from TeamForge 6.2

```
cd /opt/collabnet/teamforge/add-ons/teamforge_cli_server
./install --uninstall
zypper remove teamforge_cli_server
```

7. Install the following application packages.


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

```
zypper install teamforge-app teamforge-etl
```

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

```
zypper install teamforge-codesearch
```

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

 **Note:** Back up your `site-options.conf` file before making any changes.

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

- a) Update the host name and domain name, if required.

```
HOST_localhost=app etl indexer
```

```
HOST_my.db.host=database datamart
```

```
HOST_my.scm.host=subversion cvs gerrit
```

```
DOMAIN_localhost=my.app.domain.com
```

Configure the following settings if you are installing Black Duck Code Sight.

```
HOST_my.localhost=app etl indexer codesearch
```

- b) Configure the `JAVA_HOME` token for TeamForge.

```
JAVA_HOME=/usr/java/jdk1.7.0_40
```

- c) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

|   |                   |
|---|-------------------|
| • | SSL=on            |
| • | SSL_CERT_FILE=    |
| • | SSL_KEY_FILE=     |
| • | SSL_CA_CERT_FILE= |
| • | SSL_CHAIN_FILE=   |

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- 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`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- f) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to `$auto$`, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- DATABASE\_PASSWORD
- DATABASE\_READ\_ONLY\_PASSWORD
- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD
- BDCS\_ADMIN\_PASSWORD
- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


- g) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any Alphanumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX=<A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`.

 **Important:** The password-related tokens cannot contain the following characters: \$<>/\ ' " ` in the `site-options.conf` file.

- h) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and PostgreSQL automatically.
- i) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

```
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
```

- 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
```

 **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 username add this token:

```
BDCS_ADMIN_USERNAME=<sysadmin>
```

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) To enable the history protection feature of TeamForge Git integration, set the `GERRIT_FORCE_HISTORY_PROTECTION=true`. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
- m) If you have LDAP set up for external authentication, you must set the `"REQUIRE_USER_PASSWORD_CHANGE"` site options token to false.
- n) If you are installing TeamForge through disconnected media, set the token `HELP_AVAILABILITY=local`.
- o) Save the `site-options.conf` file.

9. Run the following command to remove the pagespeed cache.

```
cd /opt/collabnet/teamforge/cache
rm -rf pagespeed
```

**10. Back up the old TeamForge runtime directory.**


```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

**11. Recreate the runtime environment.**

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./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.


**Do the following on the database server - my.db.host**

**13. Configure your TeamForge 7.1 installation repository.** See [TeamForge installation repository configuration for SUSE](#)

**14. If the PostgreSQL database is running locally, stop the PostgreSQL service.**

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

**15. Uninstall the PostgreSQL RPMs.**


 **Note:** When prompted, you must select the option to remove all the dependent packages.

```
zypper remove postgresql-libs postgresql-docs postgresql-server postgresql
mv /var/lib/pgsql /var/lib/pgsql_9.0
```

**16. Install the TeamForge database packages.**

```
zypper install teamforge-database
```

**17. Copy the site-options.conf file from my.app.host and modify the token settings.**

 **Note:** If you choose to use the old site-options.conf file, don't forget to copy the *AUTO\_DATA* token from the application server.

```
HOST_localhost=database datamart
```

```
DOMAIN_localhost=my.db.domain.com
```

```
HOST_my.app.host=app etl indexer
```

```
HOST_my.scm.host=subversion cvs gerrit
```

**18. Back up the old TeamForge runtime directory.**

```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```


**19. Recreate the runtime environment.**

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

**20. Restore your site data.**

a) Reload the PostgreSQL data.

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

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


```
su - postgres
/usr/bin/psql -p <reports_database_port> < /tmp/dbbackup_dir/
teamforge_reporting_data_backup.dmp
exit
```

21. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

**Do this on application server - my.app.host**

22. Convert your site data to work with TeamForge 7.1.


 **Tip:** Before you kick off the data migration, use the `/etc/init.d/collabnet status` command to make sure the Jboss, Tomcat and Tomcats 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 7.1 site.

23. Run the following script to upgrade the [index to Lucene 4.x format](#).

 **Note:** You must back up the existing search index directory before running this script. Refer to the `SEARCH_INDEX_LOCATION` token in the `runtime-options.conf` file to know the search index directory location.


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

24. Run the following script to upgrade the Subversion working copies.

```
/opt/collabnet/teamforge/runtime/scripts/svn-upgrade-working-copies.sh
```

25. 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.

26. If you are upgrading from TeamForge 7.0, run the `post-install.py` script.


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

27. If you are upgrading from TeamForge 6.2 or earlier versions and the CLI reports are already configured, follow these steps:

a) Back up the CLI reports.

```
mkdir /root/backup
cd /root/backup
backup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> find . -name .svn | xargs rm -rf
```

b) Remove the `cli/reports` folder from the branding repository.

 **Note:** Do a SVN delete to remove this folder from the repository.

```
mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> svn delete reports
cli> svn commit -m "To delete the old CLI reports folder"
```

c) Run the `post-install.py` script.

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

d) Commit the backup 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```
mkdir /root/restore
cd /root/restore
restore> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
mkdir custom-reports
cp -avx /root/backup/pkg custom-reports
svn add custom-reports
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>
```

When prompted, enter the user name (`<admin_username>`) and password (`<admin_password>`)

```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```


28. Install the Black Duck Code Sight license on the server where Black Duck Code Sight is installed. For more information see [these instructions](#).

29. 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.

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

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

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

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL path of the SCM integration server, for example, `"http://myint.box.net/svn/repos"`, where `myint.box` is the server with the SCM integration server.

In addition, if you add a new integration server at some point later, you must run this `svn_cache.sh` script, (after creating the new integration server), on the TeamForge application server.



31. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

32. Run the following initial load jobs (ETL).

- 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 325.


### Do the following on the SCM server - my.scm.host

If your TeamForge setup includes source control running on its own server, you'll have to upgrade that server as well as the main TeamForge application server.

33. Stop TeamForge.

```
/etc/init.d/apache2 stop
/etc/init.d/collabnet stop tomcat
```

34. Back up your SCM data.

 **Tip:** `/tmp` in the following step is just an example. You can use any directory or partition that you prefer.

- a) Make an archive with the following data directories.

| Directory             | Contents  |
|-----------------------|---|
| <code>/svnroot</code> | Subversion source code repositories                     |
| <code>/cvsroot</code> | CVS source code repositories (not present in all sites) |
| <code>/gitroot</code> | Git source code repositories                            |

```
mkdir -p /tmp/scmbackup_dir/gerrit
cp -Rpfv /svnroot /cvsroot /tmp/scmbackup_dir
```

- b) Back up your Git database and make a dump of your Git database. You have to do a PostgreSQL dump because we are upgrading the PostgreSQL application as part of this upgrade.

```
su - postgres
/usr/bin/pg_dumpall > /var/lib/pgsql/backups/git_data_backup.dmp
exit
```

Copy the database backup to the backup directory.

```
cp /var/lib/pgsql/backups/git_data_backup.dmp /tmp/scmbackup_dir/
cp -Rpfv /gitroot /tmp/scmbackup_dir
cp -Rpfv /opt/collabnet/gerrit/etc /opt/collabnet/gerrit/.ssh /tmp/scmbackup_dir/gerrit
```

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

35. Move the collabnet repository of the older version of TeamForge.

```
mv /etc/zypp/repos.d/collabnet-7.0.0.x.repo /etc/zypp/repos.d/
collabnet-7.0.0.x.repo.cn_backup
```

 **Note:** Replace "x" with the appropriate patch release number if applicable.


36. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

37. If Git is enabled and the PostgreSQL database is running locally, follow these steps.

a) Stop the PostgreSQL service.

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

b) Uninstall the PostgreSQL RPMs.

 **Note:** When prompted, you must select the option to remove all the dependent packages.

```
zypper remove postgresql-libs postgresql-docs postgresql-server
postgresql
mv /var/lib/pgsql /var/lib/pgsql_9.0
```

38. Install the following application packages.


a) Install the source code component.

```
zypper install teamforge-scm
```

b) To install the Git packages, run the following command.

```
zypper install teamforge-git
```

39. Copy the master `site-options.conf` file from `my.app.host` and modify the host token settings in the `site-options.conf` file.

 **Note:** If you choose to use the old `site-options.conf` file, don't forget to copy the `AUTO_DATA` token from the application server.

```
HOST_localhost=subversion cvs gerrit
```

```
DOMAIN_localhost=my.scm.domain.com
```

```
HOST_my.app.host=app etl indexer codesearch
```

```
HOST_my.db.host=database datamart
```

Save the `site-options.conf` file.

40. Back up the old TeamForge runtime directory.

```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

41. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

42. Restore your Git data (if Git is enabled).

a) Reload the PostgreSQL data.

```
su - postgres
/usr/bin/psql < /tmp/scmbbackup_dir/git_data_backup.dmp
exit
```


43. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
```

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

#### 44. 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.

#### 45. Start the Tomcat service.

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

#### 46. If you have installed Git, integrate Gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

##### 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
```

#### 47. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.1.


##### a) Log into your site as the administrator.

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

See [Customize anything on your site](#).

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


See [Create a site-wide broadcast](#).

 **Important:** Do not delete the `teamforge-installer/7.1.0.0` directory. You will need it for future maintenance and upgrades.

#### 48. Remove the backup files after the TeamForge site is up and running as expected.

##### a) Remove the repository and the file system backup from the `/tmp/backup_dir` directory.

##### b) Remove the PostgreSQL 9.0 database dump and the file system from the `/var/lib/pgsql/9.0/backups` and `/var/lib/pgsql/9.0/data` directories respectively.

 **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](#).

To upgrade Review Board [see these instructions](#).

### Upgrade an advanced TeamForge site to TeamForge 7.1

Upgrading to TeamForge 7.1 on an advanced site can be complicated but you get more flexibility and control.

If there is any doubt about what kind of site you are working with, see [Is my TeamForge site "dedicate"d or "advanced"?](#) on page 253

## Upgrade to TeamForge 7.1 with Oracle Database services on a separate server

In this procedure, we'll assume that you are upgrading on the same server where your existing TeamForge 7.0 site is running.

In this option, the following services run on the TeamForge Application Server (We call this my.app.host).


- TeamForge Application Server
- Black Duck Code Sight Server
- ETL Server
- Search Server (Indexer)

The following service runs on the Database server (Oracle). (We call this my.db.host)

- Database Server (Operational DB and Reports DB)

The following service runs on the SCM server. (We call this my.scm.host)

- SCM Integration Server (Subversion and CVS)

 **Note:** For the ETL service to run as expected in a distributed TeamForge installation, all servers must have the same time zone.

**Log in to the servers as root.**

**Do this on the Oracle database server - my.db.host**

1. Make a dump file of your site database.


To back up the Oracle database, follow the [Oracle backup procedure](#).

**Do the following on the TeamForge Application Server (We call this my.app.host)**

2. Stop the Apache server and the TeamForge application server.

```
/etc/init.d/apache2 stop
/etc/init.d/collabnet stop
```

3. Back up the file system data.


 **Tip:** /tmp/backup\_dir is just an example. You can use any directory or partition you prefer to store your backup files.

- a) Make an archive 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 -p /tmp/backup_dir
cp -Rpfv /sf-svnroot /opt/collabnet/teamforge/var /tmp/backup_dir
```

4. If the SCM services are running on the TeamForge Application Server (my.app.host), do the following.

 **Tip:** /tmp/backup\_dir is just an example. You can use any directory or partition you prefer to store your backup files.

- a) Back up your SCM data.
- b) 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). |

```
cp -Rpfv /svnroot /cvsroot /tmp/backup_dir
```

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


5. Move the collabnet repository of the older version of TeamForge.

```
mv /etc/zypp/repos.d/collabnet-7.0.0.x.repo /etc/zypp/repos.d/
collabnet-7.0.0.x.repo.cn_backup
```

 **Note:** Replace "x" with the appropriate patch release number if applicable.

6. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

7. Uninstall the TeamForge CLI add-on (if it is already installed).

 **Note:** Skip this step if you are upgrading from TeamForge 6.2

```
cd /opt/collabnet/teamforge/add-ons/teamforge_cli_server
./install --uninstall
zypper remove teamforge_cli_server
```

8. Install the following application packages.

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

```
zypper install teamforge-app teamforge-etl
```


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

```
zypper install teamforge-codesearch
```

- c) If the SCM services are running on my.app.host, install the Source Code component of the TeamForge application.

```
zypper install teamforge-scm
```

9. Update the `site-options.conf` file.

 **Important:** Back up your `site-options.conf` file before making any changes.

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

- a) Update the host name and domain name, if required.

```
HOST_localhost=app etl indexer
```

```
HOST_my.db.host=database datamart
```

```
HOST_my.scm.host=subversion cvs
```

```
DOMAIN_localhost=my.app.domain.com
```

- b) Add "codesearch" to `Host_localhost` if you are installing Black Duck Code Sight.

```
HOST_localhost=app etl indexer codesearch
```

- c) Configure the `JAVA_HOME` token for TeamForge.

```
JAVA_HOME=/usr/java/jdk1.7.0_40
```

- d) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

```

• SSL=on
• SSL_CERT_FILE=
• SSL_KEY_FILE=
• SSL_CA_CERT_FILE=
• SSL_CHAIN_FILE=

```

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- e) 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

```

- f) If the token `REQUIRE_PASSWORD_SECURITY` is enabled, then set a value for the token, `PASSWORD_CONTROL_EFFECTIVE_DATE`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.


- g) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the [OBFUSCATION\\_PREFIX](#) on page 405, set `OBFUSCATION_PREFIX=<A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`).

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' "`` in the `site-options.conf` file.

- h) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

```

USERS_WITH_NO_EXPIRY_PASSWORD=admin,nobody,system,scmviewer

```


- 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.

 **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 username add this
token:
BDCS_ADMIN_USERNAME=<sysadmin>
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) If you have LDAP set up for external authentication, you must set the `"REQUIRE_USER_PASSWORD_CHANGE"` site options token to false.
  - l) If you are installing TeamForge through disconnected media, set the token `HELP_AVAILABILITY=local`.
  - m) Save the `site-options.conf` file.
10. Download the corresponding version of the Oracle client from <http://www.oracle.com/technetwork/database/features/instant-client/index-097480.html>

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

11. Run the following command to remove the pagespeed cache.

```
cd /opt/collabnet/teamforge/cache
rm -rf pagespeed
```

12. Back up the old TeamForge runtime directory.

```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

13. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

14. 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.

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
ListenBackLog     2048
</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
```

18. Run the following script to upgrade the *index to Lucene 4.x format*.

 **Note:** You must back up the existing search index directory before running this script. Refer to the `SEARCH_INDEX_LOCATION` token in the `runtime-options.conf` file to know the search index directory location.

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

19. Run the following script to upgrade the Subversion working copies.

```
/opt/collabnet/teamforge/runtime/scripts/svn-upgrade-working-copies.sh
```

20. 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.

21. If you are upgrading from TeamForge 7.0, run the `post-install.py` script.

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


22. If you are upgrading from TeamForge 6.2 or earlier versions and the CLI reports are already configured, follow these steps:

a) Back up the CLI reports.

```
mkdir /root/backup
cd /root/backup
backup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/reports/pkg
backup> find . -name .svn | xargs rm -rf
```



- b) Remove the `cli/reports` folder from the branding repository.

 **Note:** Do a SVN delete to remove this folder from the repository.

```
mkdir /tmp/cleanup
cd /tmp/cleanup
cleanup> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
cli> svn delete reports
cli> svn commit -m "To delete the old CLI reports folder"
```

- c) Run the `post-install.py` script.

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

- d) Commit the backup 'pkg' folder which is available in this location `branding/cli/custom-reports/`

```
mkdir /root/restore
cd /root/restore
restore> svn checkout --username XXXXX https://<hostname>/svn/
repository-internal/branding/cli/
cd cli
mkdir custom-reports
cp -avx /root/backup/pkg custom-reports
svn add custom-reports
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>
```

When prompted, enter the user name (<admin\_username>) and password (<admin\_password>)


```
ctf> whois admin ids cliserver set command run system/once/migrate.ctf
ctf> \q
```

23. 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.

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

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

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

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
./svn_cache.sh <Repository Base URL Path of the SCM Integration Server>
```

Provide a repository base URL path of the SCM integration server, for example, "`http://myint.box.net/svn/repos`", where `myint.box` is the server with the SCM integration server.

In addition, if you add a new integration server at some point later, you must run this `svn_cache.sh` script, (after creating the new integration server), on the TeamForge application server.

25. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

**26. Run the following initial load jobs (ETL).**

- 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 325.


**Do this on the SCM server (my.scm.host)**

If your TeamForge setup has the Source Control running on its own server, you'll have to upgrade that server as well.

**27. Stop TeamForge.**

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

**28. Back up your SCM data.**

 **Tip:** /tmp in the following step is just an example. You can use any directory or partition that you prefer.

- a) Make an archive with the following data directories.

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

```
cp -Rpfv /svnroot /cvsroot /tmp/scmbackup_dir
```

- b) Back up your SSH keys, if any.

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

**29. Move the collabnet repository of the older version of TeamForge.**

```
mv /etc/zypp/repos.d/collabnet-7.0.0.x.repo /etc/zypp/repos.d/
collabnet-7.0.0.x.repo.cn_backup
```

 **Note:** Replace "x" with the appropriate patch release number if applicable.

**30. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)**

**31. Install the following application package.**

- a) Install the source code component.

```
zypper install teamforge-scm
```

32. Copy the master `site-options.conf` file from `my.app.host` and modify the host token settings in the `site-options.conf` file.

```
HOST_localhost=subversion cvs
```

```
DOMAIN_localhost=my.scm.domain.com
```

```
HOST_my.app.host=app etl indexer codesearch
```

```
HOST_my.db.host=database datamart
```

Save the `site-options.conf` file.

33. Back up the old TeamForge runtime directory.


```
mv /opt/collabnet/teamforge/runtime /opt/collabnet/teamforge/runtime.old
```

34. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

35. 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.

36. Start the Tomcat service.

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

37. Apply the finishing touches and make sure everything is running smoothly after upgrading to TeamForge 7.1.

- Log into your site as the administrator.
- If your site has custom branding, verify that your branding changes still work as intended.  
See [Customize anything on your site](#).
- Let your site's users know they've been upgraded.  
See [Create a site-wide broadcast](#).

 **Important:** Do not delete the `teamforge-installer/7.1.0.0` directory. You will need it for future maintenance and upgrades.

## Upgrade to CollabNet TeamForge 7.1 on a virtual machine

You can upgrade to CollabNet TeamForge 7.1 from a VMware installation of CollabNet TeamForge 7.0.


 **Note:**

- To upgrade from any version earlier than 7.0 (including CollabNet SourceForge Enterprise 5.x or any version of SourceForge Enterprise Edition), first [upgrade your site to TeamForge 7.0](#) and start it up. Then follow these steps for upgrading to CollabNet TeamForge 7.1.
- When you upgrade to TeamForge 7.1, 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 7.1 for VMware Player or VMware ESXi](#)
- [Configure CollabNet TeamForge on VMware Player](#)

- 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 7.1 from your CollabNet Support representative and unzip it in the /tmp directory.

```
unzip updater-teamforge-vmware-7.1.0.0.zip
```

3. Back up your TeamForge 7.0 site.

See [Back up CollabNet TeamForge data](#) on page 262 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-7.1.0.0
./updater-7_0-7_1.sh
```

The upgrade can take several minutes to complete, depending on system resources.

 **Important:** Type Y when prompted.

### 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.

6. After logging on to TeamForge, click the **Projects** tab.
7. Click the **Look** project.
8. Click **Project Admin** in the project navigation bar.
9. Click **Project Toolbar** from the **Project Admin Menu**.
10. Select the **Linked Applications** tab.
11. Select the **CLI Jobs** linked application's check box and click **Edit**.
12. Modify the URL: Replace the "localhost" with the hostname or IP address of the computer that hosts the TeamForge application.
13. Click **Save**.

### Troubleshooting: VMware installer failed to install new TeamForge version

When your TeamForge 7.1 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-7.1.0.0.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/7.1.0.0
./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 7.1 installation

When your TeamForge 7.1 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/7.1.0.0/conf/site-options.conf
```

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

#### 2. Look for the `DEDICATED_INSTALL` variable.

- If `DEDICATED_INSTALL` is set to true, the TeamForge you have a dedicated installation, with the default configuration and minimal user intervention.
  - If `DEDICATED_INSTALL` is set to false, or is not present, you have an advanced installation, with customizations appropriate to this particular site's conditions and use patterns.
3. As you work through the instructions for upgrading or patching your site, watch for notes like this one:

 **Note:** If you are working with a *dedicated* TeamForge installation, you can skip this. See [Is my TeamForge site "dedicate"d or "advanced"?](#) on page 253


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 7.1

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 7.1

The following procedure is an outline of what you need to when upgrading from TeamForge 6.1.1 (no reporting) to TeamForge 7.1 (with reporting).

1. Stop TeamForge.
2. Back up your site data.
3. Install the TeamForge 7.1 build, making sure that you change the 6.1 site-options for the `jdk` and `MaxPermSize` requirements in 7.1.

 **Note:** Tokens related to reporting should not be enabled.

4. Migrate the data to TeamForge 7.1.
5. Stop all services. [Enable reporting-related tokens in site-options.conf](#) and recreate the runtime.
6. Start the database if it is not running.
7. From the `runtime/scripts` directory, run the `bootstrap-reporting-data.py` script.
8. 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 275.

9. Start the TeamForge 7.1 site and make sure that reporting is enabled.


## Troubleshooting: Upgrade PostgreSQL manually

The TeamForge upgrade utility upgrades PostgreSQL for you automatically when you run the `prepare-environment.sh` script. However, when you upgrade to TeamForge 6.1 from TeamForge 5.4, patch 1, it is possible for the automatic PostgreSQL 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/dumpspath/<name>.dmp
```

## Install a different build of TeamForge 7.1

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 7.1](#) on page 111.

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/7.1.0.0
```

 **Note:** Make sure the token `SCM_DEFAULT_SHARED_SECRET` is present in the `site-options.conf`.

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

4. Start TeamForge.

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

**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.

## Maintain your TeamForge 7.1 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 PostgreSQL using PGTurant

PGTurant, a wrapper for PostgreSQL's `pg_upgrade` (formerly known as `pg_migrator`), upgrades data stored in older versions of `pgsql` data files to a later `pgsql` major version, without data dump and reload, which otherwise is required for upgrades to major `pgsql` versions.

PGTurant is a multi-version PostgreSQL database system migrator that is capable of upgrading the PostgreSQL database file system to higher versions. As all the required PostgreSQL server, client and contrib RPMs are self-contained, customers need not worry about the required dependency packages while upgrading to higher PostgreSQL versions.

Though this tool was primarily intended for use by TeamForge applications during upgrades, it can also be used for any general purpose PostgreSQL upgrades provided the following prerequisites are met.

- Supported OS: 64-bit RHEL and CentOS (versions 6.x and 7.x)



**Important:** You must have RHEL/CentOS 6.5 or later to run PGTurant. Upgrade the operating system packages (`yum upgrade`) before running PGTurant on sites running on RHEL/CentOS versions 6.4 or earlier.

- Pre-configured OS and CollabNet yum repository
- Python 2.6 or later
- A shell with 'root' access

The following table lists the supported PGSQL version for various TeamForge versions.

TeamForge version	Supported PostgreSQL version
6.2 (including 6.2 Patch 1)	9.0.13
7.0 (including 7.0 Patch 1)	9.0.13
7.1	9.2.4

If you are in PostgreSQL 9.0, you can upgrade to PostgreSQL 9.2.

1. Starting TeamForge 6.2, PGTurant RPMs are distributed via CollabNet TeamForge's yum repository. Check your yum repository for PGTurant RPMs.
  - a) Log on to your server as root user.
  - b) Check for PGTurant package information.

```
yum clean all
yum info pgturant
```



If you see PGTurant package information, your yum repository is fine. Otherwise, contact [CollabNet Support](#) for resolution.

## 2. Install PGTurant RPM.

```
yum install pgturant -y
```

PGTurant files are installed in the `/opt/collabnet/pgturant/` directory.

### PGTurant command line options and usage examples

Option	Description
-D --debug	Use this option to enable debug mode. Enabling the debug mode displays all the back-end debug messages on the console.
-s --srcdir	Use this option to pass the PostgreSQL source database directory as a command line parameter. Suppose you want to upgrade your PostgreSQL data from v9.0 to v9.3 and that your v9.0 data directory is <code>/var/lib/pgsql/9.0/data</code> . Syntax for this option would be: <code>-s /var/lib/pgsql/9.0/data</code> .
-d --destdir	Use this option to pass the PostgreSQL destination database directory as a command line parameter. Suppose you want to upgrade your PostgreSQL data from v9.0 to v9.3. Syntax for this option would be: <code>-d /var/lib/pgsql/9.3</code> . By default, the destination directory is assumed as <code>/var/lib/pgsql/&lt;upgrade version&gt;</code> , where <code>&lt;upgrade version&gt;</code> is specified in <code>-u</code> option.
-u --upgrade	Use this option to pass the destination PostgreSQL version as a command line parameter so that the database files are upgraded to that specific PostgreSQL version supported by PGTurant.
-S --setup	Use this option to install the required setup environment for PGTurant to be able to upgrade data to the desired PostgreSQL version.
-c --check	Use this option to just check the database clusters and not change the database content. You can think of it as the dry-run mode.
-m --migrate	Use this option to do the actual migration of database system to new PostgreSQL version as specified in the <code>-u</code> option.
-h --help	Use this option to display PGTurant help information.

#### Example 1: Upgrade PostgreSQL database files from v9.0 to v9.2 - dry-run mode

```
cd /opt/collabnet/pgturant/bin
```

```
./pgturant -s /var/lib/pgsqli/9.0/data -d /var/lib/pgsqli/9.2 -u
9.2 -c
```

**Example 2: Upgrade PostgreSQL database files from v9.0 to v9.2 - upgrade database**

```
cd /opt/collabnet/pgturant/bin
./pgturant -s /var/lib/pgsqli/9.0/data -d /var/lib/pgsqli/9.2 -u
9.2 -m
```

**Example 3: Upgrade PostgreSQL database files from v9.0 to v9.2 - dry-run and upgrade database**

```
cd /opt/collabnet/pgturant/bin
./pgturant -s /var/lib/pgsqli/9.0/data -d /var/lib/pgsqli/9.2 -u
9.2 -c -m
```

## 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.

```
/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 search index directory. Copy or transfer the search index 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 /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -d /opt/collabnet/teamforge
```

5. Start TeamForge.

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

**Do this on the TeamForge Indexer server.**

6. Set up the server with an operating system supported by TeamForge 7.1

7. Copy the following repository to your production server.

```
vi /etc/yum.repos.d/collabnet-7.1.0.0.repo
```

```
[CollabNet]
name=collabnet
baseurl=http://packages.collab.net/7.1.0.0/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 the Indexer server.
- The Indexer server should have the required permissions to access the NFS volume of the application 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 application server.

```
cd /opt/collabnet/teamforge/var/
tar -zxvf /tmp/searchIndex.tgz
```

12. Run the installer.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -d /opt/collabnet/teamforge
```

13. Start the TeamForge Indexer.

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

- a) Run the following command and ensure that application filestorage directory is exported to the indexer server.

```
mount -l
```


- b) Run the following command and ensure that the Indexer services are up and running.

```
/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 260.

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 7.1](#) 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/7.1.0.0/conf/site-options.conf
```

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

2. In the `site-options.conf` file, comment out these variables:

- `USE_EXTERNAL_USER_AUTHENTICATION`
- `LOGIN_CONFIG_XML_FILE`
- `MINIMUM_PASSWORD_LENGTH`

3. Restart the runtime environment.

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

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

### Back up CollabNet TeamForge data

Save a copy of your TeamForge site's data to a location from where you can quickly retrieve it to your TeamForge 7.1 site.

- If you are upgrading by installing TeamForge 7.1 on new hardware, then you'll need the backed-up site data to complete the upgrade.
- If you are upgrading your site on the same hardware, then you won't need the backed-up data but you should create it anyway, as a precaution.

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

```

/etc/init.d/httpd stop
/etc/init.d/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 Lucene upgrade for search functionality, upgrading to TeamForge 7.1 requires a complete re-index of the site. This could take several hours, and the index data could double in size.

## Permit big file uploads


When many users store very large files on your site, you may sometimes notice a slowdown in your site's performance. You can reduce the impact of such a use pattern by telling TeamForge not to index files larger than a certain size.

 **Note:** It's also a good idea to let your users know that the Documents tool in TeamForge is not designed primarily as a storage device. As a best practice, upload documents to make them available for collaboration, not for backup or long-term storage.


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

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

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

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

2. Add the `SEARCH_MAX_FILE_SIZE` parameter and give it a value equal to the maximum size (in bytes) of files to be indexed.

 **Note:** The default value is 10M. With this value, files larger than 10M are not indexed.

A value of zero or less specifies that there is no limit, which is the same as the default behavior without the variable.

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

### Allow searching of archive files

By default, users can't search the content of archive files uploaded to TeamForge, such as `zip`, `tar`, or `docx` files. If your users need it, you can provide this ability.

Microsoft Office 2007 files, such as files with the `.docx` extension, are archive files. By default their content is not indexed and does not show up in search results. However, information that TeamForge maintains about those documents, such as title, author, description and version, does appear in search results.

If you permit archive searching, watch for performance slowdowns associated with the larger volume of indexing that TeamForge is doing. Depending on your site members' use patterns, the performance cost may or may not be acceptable to your users.

- 👉 **Note:** It's also a good idea to let your users know that the Documents tool in TeamForge is not designed primarily as a storage device. As a best practice, upload documents to make them available for collaboration, not for backup or long-term storage.

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

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

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

- 👉 **Note:** `vi` is an example. Any \*nix text editor will work.

2. Add the `SEARCH_SUPPRESS_ARCHIVE_SUB_DOCS` parameter and set it to false. This enables files inside archives (such as `.zip`, `.gz`, or `.tar`) to be indexed for search.
3. Review the variables you've changed, then save the `site-options.conf` file.

### Limit the size of message attachments

To avoid overtaxing your mail server or your storage volume, you may want to set a ceiling on the size of the attachments that users can send to a forum via email.

When a user sends an attachment that is larger than the limit, the message is rejected and the user gets an email from the Site Administrator explaining that the attachment exceeded the limit.

- 👉 **Tip:** Before imposing a file attachment size limit, it's a good idea to point your users to better ways of collaborating around large files. Consider suggesting source code repositories, backup systems, or other appropriate solutions.

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

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

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

- 👉 **Note:** `vi` is an example. Any \*nix text editor will work.

2. Set the value of the `DISCUSSION_MAX_ATTACHMENT_SIZE` variable to a reasonable level. For example, if your users are given to using Microsoft Word documents on the site, you might set `DISCUSSION_MAX_ATTACHMENT_SIZE` to 10 MB, and increase the value by two or three MB at a time if users need more headroom.
3. Review the variables you've changed, then save the `site-options.conf` file.

### Limit the size of document attachments


When many users store very large documents on your site, you may sometimes notice a slowdown in your site's performance. You can reduce the impact of such a use pattern by telling TeamForge not to attach documents larger than a certain size.

- 👉 **Note:** It's also a good idea to let your users know that the Documents tool in TeamForge is not designed primarily as a storage device. As a best practice, upload documents to make them available for collaboration, not for backup or long-term storage.

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

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

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

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

2. Add the `DOCUMENT_MAX_FILE_UPLOAD_SIZE` parameter and give it a value equal to the maximum size (in megabytes) of documents to be uploaded.
3. Review the variables you've changed, then save the `site-options.conf` file.

### Who can post to discussions by email?


To help reduce the risk of spam or other mischief, you may need to limit the users who can post to discussion forums by email.

To leverage the advantages of community collaboration, you should keep your forums as open as you can. However, some sites require tighter control over who can participate in discussions. TeamForge enables you to balance openness against privacy along a spectrum of choices.

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

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

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

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

2. Add the `DISCUSSION_EMAIL_POSTING` variable, and give it the value that reflects the degree of openness your site's discussion forums require.

Use one of these values:

Value	Description
0	Allow only forum admins.
1	Users with roles and permissions.
4	All logged in users.
5	Allow known email addresses only.
6	Allow all site users and guests.

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

The value you set here determines the maximum degree of openness to email posting for all projects on your site. For example, consider a site where project members can post by email (level 3). For a project that requires extra security, the project administrator can choose to accept email only from users with the appropriate role (level 1). However, a project owner cannot accept email posts from a less restrictive category of users, such as all users who are logged in (level 4).

### Who can monitor discussions?


You may need to limit the users who can monitor discussion forums.

To leverage the advantages of community collaboration, you should keep your forums as open as you can. However, some sites require tighter control over how users keep track of discussions. TeamForge enables you to balance openness against privacy along a spectrum of choices.

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

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

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

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

2. Add the `DISCUSSION_EMAIL_MONITORING` variable, and give it the value that reflects who you want to get updates on discussions via email.

Use one of these values:

Value	Description
0	Allow only forum admins.
1	Users with role permissions.
4	All logged in users.
5	Allow all site users and guests.

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

The value you set here determines the maximum degree of openness to monitoring discussions for all projects on your site. For example, consider a site where project members can monitor discussions (level 3). For a project that requires extra security, the project administrator can dictate that only users with the appropriate role can monitor discussions (level 1). However, a project owner cannot allow monitoring for a less restrictive category of users, such as all users who are logged in (level 4).


### Reduce discussion spam

You can filter out some kinds of spam from your project's discussion forums.

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

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

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

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

2. Add one or more MIME types to the Reject MIME types filter.

The presence of any of these MIME types in an incoming message (via email) will cause its rejection with appropriate notification to the posting user.

For example: `DISCUSSION_REJECT_MIME_TYPES=application/pdf,text/xml`

3. Add one or more MIME types to the Drop MIME types filter.

The presence of any of these MIME types in an incoming message (via email) will cause its attachment to be deleted with appropriate notification to the posting user.

For example: `DISCUSSION_DROP_MIME_TYPES=image/jpeg,image/jpg,text/xml`

4. Add one or more header names to the remove headers filter.

If an incoming email posting contains any of these headers, they will be quietly removed from the message before it is archived and before subscribers are notified.

For example: `DISCUSSION_REMOVE_HEADERS=precedence,x-no-archive,Return-Path`

5. Add one or more header names to reject headers filter to be rejected or moderated (if discussion is moderated).

Use regular expressions, each regular expression must match an entire header. The match of any of these headers in an incoming message (via email) will cause its rejection with appropriate notification to the posting user.

For example: `DISCUSSION_REJECT_HEADERS=(?s).*headername1:value2.*,(?s).*name2:value2.*`

6. Add one or more entries for Reject content filter.

Use regular expressions, each regular expression must match an entire entry. The match of any of these entries in discussion body and subject of an incoming message (via email) will cause its rejection with appropriate notification to the posting user.

For example: `DISCUSSION_REJECT_CONTENT=(?s).*word.*,(?s).*spam.*`

 **Note:** The content entry is a case sensitive.

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

8. Restart the site.

### Install project templates 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/7.1.0.0
./install-project-templates.py -V
```

Use a site administrator user name and password. For a new site, these are `admin` and `admin`.

 **Tip:** On some servers, it may take a few seconds for the SOAP server to be ready after installation. If `install-project-templates.py` returns an error, try waiting briefly and then running it again.


### Install project templates using the installer

With TeamForge 7.1, 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/7.1.0.0/conf/site-options.conf
```

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

2. Change the value of the `SHOW_PROJECT_TEMPALTE_DETAIL_TO_NON_SITEADMINS` variable to true.

 **Note:** Use the parameter name as given, including the typo.

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

#### 4. Recreate the runtime environment.

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

#### Provide a Perforce source control server

Enabling your users to integrate Perforce repositories into their TeamForge projects requires some extra configuration.


1. If you are adding Perforce support to an existing site, back up your site's data first. (If you are adding Perforce as part of installing a new site, skip this.)

See [Back up CollabNet TeamForge data](#) on page 284.


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/7.1.0.0/conf/site-options.conf
```

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

4. Add these variables to your `site-options.conf` file, changing the values as appropriate:

 **Note:** In case of a multibox setup, modify 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 285.


## 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, include the token `DATABASE_SSL=on`.
- b) If the datamart database is running on a separate server, include 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/7.1.0.0
./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. If your TeamForge site is running in advanced mode, stop the external services by running the following commands.

- Stop the Apache server.

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

- If the TeamForge instance uses PostgreSQL database, stop the PostgreSQL.

```
/etc/init.d/postgresql-9.2 stop
```

2. Stop TeamForge.

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

3. 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
```

4. Add the following site option token in the TeamForge application server.

```
SELINUX_SETUP=true
```

5. 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
```

#### 6. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

#### 7. 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.

#### 8. 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-9.2 start
```

#### 9. Start TeamForge.

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

### 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/7.1.0.0/conf/site-options.conf
```

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


#### 3. Set the options to enable SSL for the site.

- a) Set the `SSL` variable to on.
- b) Set the `SSL_CERT_FILE` variable to the location of the file that contains your site's SSL certificates.

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

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

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

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

4. In the `site-options.conf` file, make sure the value of the `DOMAIN_localhost` variable matches that of your SSL certificate.
5. Rename the `/etc/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 all TeamForge services.

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

## 8. Run the installer.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./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/collabnet start all
```

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.

Since your site is configured to run in the SSL mode (`http` to `https`), there is a change in the URI scheme. Run the following post installation scripts that are applicable to make your TeamForge integration components to function seamlessly.

Run the TeamForge `post-install.py` script.

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

Run the Git `post-install.py` script if you have installed Git integration.

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

If you have integrated Black Duck Code Sight, run the Black Duck Code Sight `post-install.sh` script.

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


## Set up SSL for your TeamForge site on RedHat

To force all TeamForge traffic to use SSL encryption (HTTPS), state that preference in your 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/7.1.0.0/conf/site-options.conf
```

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

### 2. 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.

### 3. In the `site-options.conf` file, make sure the value of the `DOMAIN_localhost` variable matches that of your SSL certificate.

### 4. Rename the `/etc/httpd/conf.d/ssl.conf` file to `/etc/httpd/conf.d/ssl.conf.old`, if it exists.

### 5. Stop TeamForge.

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

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

#### 6. Run the installer.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

#### 7. 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.

#### 8. Rename the `/etc/httpd/conf/httpd.conf.cn_new` file to `httpd.conf`, if it exists.

#### 9. Start TeamForge.

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

A new Apache configuration file is created with the information you provided in the `site-options.conf` file. The new file is named `httpd.conf.cn_new`. It contains `VirtualHost` sections for port 80 and port 443. All port 80 requests are redirected to port 443.

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

### Generate SSL certificates

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


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

- Purchase a SSL certificate from a certificate authority (CA). Searching the Web for "certificate authority" will present several choices.
- Generate a self-signed certificate. This option costs nothing and provides the same level of encryption as a certificate purchased from a certificate authority (CA). However, this option can be a mild annoyance to some users, because Internet Explorer (IE) issues a harmless warning each time a user visits a site that uses a self-signed certificate.

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

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


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

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

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

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


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

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

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

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

This command will prompt you for the X.509 attributes of your certificate. Give the fully qualified domain name, such as `www.example.com`, when prompted for `Common Name`.

 **Note:** Do not enter your personal name here. It is requesting a certificate for a webserver, so the `Common Name` has to match the FQDN of your website.

#### 4. Generate a self-signed certificate.

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

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


You will now have an RSA private key in `www.example.com.key`, a Certificate Signing Request in `www.example.com.csr`, and an SSL certificate in `www.example.com.crt`. The self-signed SSL certificate that you generated will be valid for 370 days.

### Prevent HTTPS cracking

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

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

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

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

2. In each section, add the following two lines:

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

3. Save the file and restart Apache.

```
apachectl restart
```

### Protect integrations with SSL


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

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

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

1. Collect server certificates from all servers.

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

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

2. Locate the Java keystore.

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

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


3. Locate the Java keytool utility.

This is `PATH_TO_JAVA/bin/keytool`

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

4. Import each server certificate into the keystore.

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


 **Note:** Any value is accepted for server in `-alias <server>`.

5. At the password prompt, use `changeit`.

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

6. Verify that all your certificates are added.

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

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

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

- a) Set `sfmain.integration.listener_ssl` to `true`.
- b) Set `sfmain.integration.listener_port` to `443`.

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

9. Restart TeamForge

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

## 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/7.1.0.0/conf/site-options.conf
```

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

2. Comment out the default `HOST_localhost=app database subversion cvs` variable, then add (or uncomment) the variable that has the reporting services enabled:

 **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 383.

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 395 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 7.1 supports Subversion 1.8.3.

1. Log in as `root` and stop all `collabnet` services.
2. Check that Subversion 1.8.3 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 7.1 supports Subversion 1.8.3.

1. Login as root and stop all collabnet services.
2. Check that Subversion 1.8.3 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 the scmviewer password

It is recommended to change the scmviewer password after installing TeamForge.

Follow these steps to change the scmviewer password:

1. Stop TeamForge.

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

2. Create an encrypted password using the [password\\_util.sh](#) on page 371

```
/opt/collabnet/teamforge/runtime/scripts/password_util.sh -encrypt
'<new_password_text>'
```

3. Set the encrypted password to the [SCM\\_USER\\_ENCRYPTED\\_PASSWORD](#) on page 416 token in the /opt/collabnet/teamforge-installer/7.1.0.0/conf/site\_options.conf file.


4. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r
```

5. Start TeamForge.

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

6. Run the Black Duck Code Sight `post-install.sh` script.

 **Important:** If the Black Duck Code Sight is running on the same server, run the `post-install.sh` script. If the Black Duck Code Sight is running on a separate server, follow steps one to four and execute the `post-install.sh` script in the Black Duck Code Sight server.

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

7. Run the following command to set the new password for Review Board.

```
cd /opt/collabnet/RBInstaller-1.4.0.0.22
python ./install.py --auth-scmuser
```

## 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/7.1.0.0/conf/site-options.conf` file, change the value of the `LOG_DIR` variable to reflect the location where you want the log files to be written.

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.

Use these instructions for TeamForge7.1 and later versions.

 **Note:** Using these instructions enables debug logging only in `vamessages.log`.


1. Edit `$RUNTIME/jboss/server/default/deploy/logging.properties` to enable logging in `vamessages.log`.

 **Note:** You need to restart the site for JBoss to pick up these changes.

2. Locate the property names and modify the values as indicated in the table:

**Table 1: Property settings to enable debugging**

Property Setting	Current Value	Modified Value	Description
<code>logger.level</code>	INFO	DEBUG	Root logger level
<code>logger.com.vasoftware.level</code>	INFO	DEBUG	Log handler for <code>VAFILE</code>
<code>handler.VAFILE.level</code>	INFO	DEBUG	Log for <code>VAFILE</code>

 **Note:** To enable James logs such as Maillet, 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/7.1.0.0/conf/site-options.conf
```

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

3. In the `site-options.conf` file, change the value of the `LOG_QUERY_TIME_THRESHOLD` variable to a value, in milliseconds, that makes sense for your environment.
4. Run the installer.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./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/7.1.0.0/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 your TeamForge 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/7.1.0.0/conf/site-options.conf
```

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


2. Set the `ETL_JOB_TRIGGER_TIME` variable to the interval at which you want ETL jobs to run. For example, a value of `0 0/15 * * * ?` will run an ETL job every 15 minutes.
3. Review the variables you've changed, then save the `site-options.conf` file.

## 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/httpd start
/etc/init.d/apache2 start
/etc/init.d/postgresql-9.0 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 by using the `db.py` script.

 **Note:** You can use the `db.py` script to restore PostgreSQL data in all setups except the dedicated database server setup where the 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 <dbname> ctfdb.dmp`
- `psql ctfdb < /tmp/backup_dir/teamforge_data_backup.dmp`

 **Note:** This example assumes that the name of the TeamForge database is `ctfdb`.

 **Tip:** It may also be necessary to restore ownership of the restored tables to the `ctfuser` database user. 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
```

### 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 7.1 with other tools

---

TeamForge 7.1 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](http://open.collab.net) and unzip it.
2. Modify these values in the `installer/install.conf` file to suit your installation environment.

Option	Description
<code>pebble.base.dir</code>	Path where you want Pebble to be installed on this host.
<code>ctf.baseurl</code>	Absolute URL of the TeamForge site that you want to associate to, such as <code>https://my.ctf.instance/</code>
<code>tomcat.port</code>	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.
<code>domain</code>	The base url to be used for Pebble, such as <code>my.ctf.instance</code> . (You don't need <code>http:</code> or <code>https:</code> here.)
<code>timezone</code>	The time zone Pebble will use to timestamp blog entries.
<code>java_home</code>	Path to a JDK 1.6.x instance.
<code>protocol</code>	<code>http</code> if SSL is not being used; <code>https</code> if SSL is being used.
<code>data.dir</code>	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 289 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 288.

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 7.1 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 7.1 with Black Duck Code Sight on a separate server on Red Hat/CentOS

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

-  **Note:** For the ETL service to run as expected in a distributed TeamForge installation, all servers must have the same time zone.

In this option, the following services run on the application server (we call this my.app.host).

- TeamForge Application Server
- Database Server (Operational DB and Reports DB)
- ETL Server
- GIT Integration Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer).

The following service runs on the Black Duck Code Sight Server. (We call this my.codesight.host )

- Black Duck Code Sight Server

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

1. Install Red Hat Enterprise Linux / CentOS 6.4 or later versions and log in as root.

- The host must be registered with the Red Hat Network if you are using Red Hat Enterprise Linux. See [Platform specification for TeamForge 7.1](#) on page 356 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 server](#) on page 7 for details.

3. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.

c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```

See [Set up SELINUX](#) on page 271 to have TeamForge to run in SELinux mode after completing the installation or upgrade.

4. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

5. 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
```

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

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

- a) Configure the HOST token.

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


```
DOMAIN_localhost=my.app.domain.com
```

```
HOST_my.codesight.domain.com=codesearch
```

- b) Configure the following settings if you are installing Git.

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

- c) Configure the database and datamart settings.

 **Note:** For more information about configuring variables, see [site-options.conf](#) on page 383

```
DATABASE_TYPE=postgresql
```

```
DATABASE_USERNAME=ctfuser
```

```
DATABASE_NAME=ctfdb
```


```
DATABASE_READ_ONLY_USER=ctfrouser
```

```
REPORTS_DATABASE_USERNAME=ctfrptuser
```

```
REPORTS_DATABASE_NAME=ctfrptdb
```

```
REPORTS_DATABASE_READ_ONLY_USER=ctfrptrouser
```

```
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

 **Note:** The database name and username values are arbitrary alphanumeric strings.

- d) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to **\$auto\$**, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- DATABASE\_PASSWORD
- DATABASE\_READ\_ONLY\_PASSWORD
- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD
- BDCS\_ADMIN\_PASSWORD
- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


- e) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`:

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- f) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

```

• SSL=on
• SSL_CERT_FILE=
• SSL_KEY_FILE=
• SSL_CA_CERT_FILE=
• SSL_CHAIN_FILE=

```

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- g) If the token `REQUIRE_PASSWORD_SECURITY` is enabled, then set a value for the token, `PASSWORD_CONTROL_EFFECTIVE_DATE`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- h) 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.
- i) 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__}
```
- j) If you have LDAP set up for external authentication, you must set the `"REQUIRE_USER_PASSWORD_CHANGE"` site options token to false.
- k) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and Postgresql automatically.
- l) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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

If you are installing Git, 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
```

- 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.

- 👉 **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 username add this token:

```
BDCS_ADMIN_USERNAME=<sysadmin>
```

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 are 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 the 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
```

- To enable the history protection feature of TeamForge Git integration, set the **`GERRIT_FORCE_HISTORY_PROTECTION=true`**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
- If you are installing TeamForge through disconnected media, set the token `HELP_AVAILABILITY=local`.
- Ensure to set the token `SELINUX_SETUP=false` temporarily in the `site-options.conf` file.

r) Save the `site-options.conf` file.

7. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```

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

```
export http_proxy=
```

**Do this on the Black Duck Code Sight Server. We'll call this `my.codesight.host`**

9. Install Red Hat Enterprise Linux / CentOS 6.4 or later versions and log in as root.

- The host must be registered with the Red Hat Network if you are using Red Hat Enterprise Linux. See [Platform specification for TeamForge 7.1](#) on page 356 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. If the TeamForge server has SELinux enabled, disable it temporarily while installing or upgrading TeamForge.

a) Verify if SELinux is running in enforcing mode.

```
getenforce
```

b) If the output of the `getenforce` command is either "Disabled" or "Permissive", SELinux is already disabled.

c) If not disabled, run the following command to disable SELinux.

```
setenforce 0
```


11. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for Red Hat/CentOS](#)

12. Run the following command to install the Black Duck Code Sight packages.

```
yum install teamforge-codesearch
```

13. Copy the `site-options.conf` file from the application server to the Code Search server in the directory `/opt/collabnet/teamforge-installer/7.1.0.0/conf`

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

 **Note:** If you choose not to use the application server's `site-options.conf` file, then don't forget to copy the value of `AUTO_DATA` token from the application server.

```
HOST_localhost=codesearch
```

```
DOMAIN_localhost=my.codesight.domain.com
```

```
HOST_my.app.domain.com=app database datamart etl indexer subversion cvs
gerrit
```

15. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V
```


**Do the following on the application server - `my.app.host`**

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

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./bootstrap-data.sh
```

17. Start TeamForge.

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

-  **Note:** Startup can take several minutes, depending on the speed of the host system configuration. 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.

18. If you have installed GIT, create the 'gitadmin' user (which is already added in the site-options token `USERS_WITH_NO_EXPIRY_PASSWORD`) with site administrator rights in the TeamForge user interface.

19. If you have installed Git, integrate Gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

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
```

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

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

21. Run the following initial load jobs (ETL).

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 325.

22. Revoke the user permissions of the database and datamart users.

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

**Do this on my.codesight.host**

23. Install the Black Duck Code Sight license on the server where Black Duck Code Sight is installed. For more information, see [Install the Black Duck Code Sight license](#) on page 302.
24. 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.

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

25. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

26. Restart the Black Duck Code Sight service.

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

### Do this on my.app.host

27. Restart the collabnet services.

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

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

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

- c) Create a sample project.

See [Create a TeamForge project](#).

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

See [Create a site-wide broadcast](#).

For specific instructions on installing Review Board, see: [Set up Review Board](#) on page 305


To run TeamForge in SELINUX enabled mode, see [Set up SELINUX](#) on page 271

### Installing TeamForge Orchestrate

To install TeamForge Orchestrate, see [TeamForge Orchestrate installation](#).

### Install TeamForge 7.1 with Black Duck Code Sight on a separate server on SUSE

In this option, we install Black Duck Code Sight on a separate server on SUSE and other services on the main application server.

 **Note:** For the ETL service to run as expected in a distributed TeamForge installation, all servers must have the same time zone.

In this option, the following services run on the application server (we call this `my.app.host`).

- TeamForge Application Server
- Database Server (Operational DB and Reports DB)
- ETL Server
- GIT Integration Server
- SCM Integration Server (Subversion and CVS)
- Search Server (Indexer).


The following service runs on the Black Duck Code Sight Server. (We call this `my.codesight.host` )

- Black Duck Code Sight Server

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



1. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.
  - See [Platform specification for TeamForge 7.1](#) on page 356 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 server](#) on page 7 for details.
3. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)
4. 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
```

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

```
vi /opt/collabnet/teamforge-installer/7.1.0.0/conf/site-options.conf
```

- a) Configure the HOST token.

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


```
DOMAIN_localhost=my.app.domain.com
```

```
HOST_my.codesight.domain.com=codesearch
```

- b) Configure the following settings if you are installing Git.

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

- c) Configure the database and datamart settings.

 **Note:** For more information about configuring variables, see [site-options.conf](#) on page 383

```
DATABASE_TYPE=postgresql
```

```
DATABASE_USERNAME=ctfuser
```

```
DATABASE_NAME=ctfdb
```


```
DATABASE_READ_ONLY_USER=ctfrouser
```

```
REPORTS_DATABASE_USERNAME=ctfrptuser
```

```
REPORTS_DATABASE_NAME=ctfrptdb
```

```
REPORTS_DATABASE_READ_ONLY_USER=ctfrptrouser
```

```
REPORTS_DATABASE_MAX_POOL_SIZE=30
```

 **Note:** The database name and username values are arbitrary alphanumeric strings.

- d) Starting TeamForge 7.1, the TeamForge installer supports automatic password creation for the following password-related `site-options.conf` tokens.

When set to `$auto$`, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file. This feature is enabled by default. You can, however, override any of the following password-related tokens with the password of your choice.

- DATABASE\_PASSWORD
- DATABASE\_READ\_ONLY\_PASSWORD
- REPORTS\_DATABASE\_PASSWORD
- REPORTS\_DATABASE\_READ\_ONLY\_PASSWORD
- ETL\_SOAP\_SHARED\_SECRET
- JAMES\_ADMIN\_PASSWORD
- BDCS\_ADMIN\_PASSWORD
- MIRROR\_DATABASE\_PASSWORD (applicable only if you are mirroring your database)


e) **Password Obfuscation**

The password obfuscation is enabled by default. As a result, all password-related tokens are encrypted in all the TeamForge configuration files.

To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.

To configure the obfuscation key, set `OBFUSCATION_KEY=<Any AlphaNumeric value with length >= 8 bytes>`. The default value of `OBFUSCATION_KEY` token is `XSJt43wN`.

To configure the `OBFUSCATION_PREFIX` on page 405, set `OBFUSCATION_PREFIX= <A value with 4 to 8 bytes length>`. The default value of `OBFUSCATION_PREFIX` is `{OBF}`).

 **Important:** The password-related tokens cannot contain the following characters: `$<>/\ ' " `` in the `site-options.conf` file.

- f) Turn on the SSL for your site by editing the relevant variables in the `site-options.conf` file. To generate the SSL certificates, see [Generate SSL certificates](#) on page 274.

```

• SSL=on
• SSL_CERT_FILE=
• SSL_KEY_FILE=
• SSL_CA_CERT_FILE=
• SSL_CHAIN_FILE=

```

 **Note:** The `SSL_CA_CERT_FILE` and `SSL_CHAIN_FILE` are optional.

- g) If the token `REQUIRE_PASSWORD_SECURITY` is enabled, then set a value for the token, `PASSWORD_CONTROL_EFFECTIVE_DATE`. The Password Control Kit (PCK) disables, deletes or expires user accounts that don't meet the password security requirements starting from the date set for the `PASSWORD_CONTROL_EFFECTIVE_DATE` token. If a date is not set, the PCK disables, deletes or expires user accounts immediately. See [PASSWORD\\_CONTROL\\_EFFECTIVE\\_DATE](#) for more information.
- h) 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.
- i) 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__}
```

- j) If you have LDAP set up for external authentication, you must set the `"REQUIRE_USER_PASSWORD_CHANGE"` site options token to false.
- k) Ensure to set the token `DEDICATED_INSTALL=true`. This makes the installation process very simple as the TeamForge installer takes care of configuring the Apache and Postgresql automatically.
- l) Set the `USERS_WITH_NO_EXPIRY_PASSWORD` token as follows:

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

If you are installing Git, 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
```

- 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.

- 👉 **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 username add this token:

```
BDCS_ADMIN_USERNAME=<sysadmin>
```

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 are 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 the 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

```

- o) To enable the history protection feature of TeamForge Git integration, set the **GERRIT\_FORCE\_HISTORY\_PROTECTION=true**. For more information see [GERRIT\\_FORCE\\_HISTORY\\_PROTECTION](#) on page 396
- p) If you are installing TeamForge through disconnected media, set the token `HELP_AVAILABILITY=local`.
- q) Save the `site-options.conf` file.

#### 6. Recreate the runtime environment.

```

cd /opt/collabnet/teamforge-installer/7.1.0.0
./install.sh -r -I -V

```

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

```
export http_proxy=
```

#### Do this on the Black Duck Code Sight Server. We'll call this `my.codesight.host`

#### 8. Install SuSE Linux Enterprise Server 11 SP2 and log in as root.

- See [Platform specification for TeamForge 7.1](#) on page 356 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.

#### 9. Check your basic networking setup.

See [Set up networking for your TeamForge server](#) on page 7 for details.


#### 10. Configure your TeamForge 7.1 installation repository. See [TeamForge installation repository configuration for SUSE](#)

#### 11. Run the following command to install the Black Duck Code Sight packages.

```
zypper install teamforge-codesearch
```

#### 12. Copy the `site-options.conf` file from the application server to the Code Search server in the directory `/opt/collabnet/teamforge-installer/7.1.0.0/conf`

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

 **Note:** If you choose not to use the application server's `site-options.conf` file, then don't forget to copy the value of `AUTO_DATA` token from the application server.

```
HOST_localhost=codesearch
```

```
DOMAIN_localhost=my.codesight.domain.com
```

```
HOST_my.app.domain.com=app database datamart etl indexer subversion cvs
gerrit
```

#### 14. Recreate the runtime environment.

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
```

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


### Do the following on the application server - my.app.host

#### 15. Set up the initial site data (bootstrap).

```
cd /opt/collabnet/teamforge-installer/7.1.0.0
./bootstrap-data.sh
```


#### 16. Start TeamForge.

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

 **Note:** Startup can take several minutes, depending on the speed of the host system configuration. 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.

#### 17. If you have installed GIT, create the 'gitadmin' user (which is already added in the site-options token *USERS\_WITH\_NO\_EXPIRY\_PASSWORD*) with site administrator rights in the TeamForge user interface.

#### 18. If you have installed Git, integrate gerrit by running the `post-install.py` script.

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

The post installation script detects the required configuration parameters. The following three parameters are not set by default. Provide a value for these parameters when prompted.

- TeamForge login name: The dedicated TeamForge site administrator account that does not expire and cannot be locked.
- TeamForge password: The password for the dedicated TeamForge site administrator account.
- Database password: The password to protect Gerrit's database from unauthorized access. Specify its value when you first run the `post-install.py` script. Make a note of the database password as you may need it later.

##### 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
```

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

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

#### 20. Run the following initial load jobs (ETL).

##### 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 325.

### Do this on my.codesight.host

21. Install the Black Duck Code Sight license on the server where Black Duck Code Sight is installed. For more information, see [Install the Black Duck Code Sight license](#) on page 302.

22. 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.

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

23. If the token `VALIDATE_SSL_CERTS` is set to "true", you must run the codesearch runtime script `trust-cert.sh` in the application server and restart the Jboss service.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch
./trust-cert.sh
/etc/init.d/collabnet -V restart jboss
```

### Do this on my.app.host

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

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

25. Restart the collabnet services.

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

### Do this on my.codesight.host

26. Restart the Black Duck Code Sight service.

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

### Do this on my.app.host

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

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

- c) Create a sample project.  
See [Create a TeamForge project](#).
- d) Write a welcome message to your site's users.  
See [Create a site-wide broadcast](#).

For specific instructions on installing Review Board, see: [Set up Review Board](#) on page 305

## 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 7.1, 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](mailto: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 6.4 and log in as root.

- The host must be registered with the Red Hat Network. See [Platform specification for TeamForge 7.1](#) on page 356 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 server](#) 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 7.1 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 7.1 disconnected installation and save it in /tmp.

OS	Installer package
Red Hat Enterprise Linux /CentOS 6.4 32-bit	CTF-Disconnected-media-7.1.0.0-xxx.el6.i386.rpm
Red Hat Enterprise Linux/CentOS 6.4 64-bit	CTF-Disconnected-media-7.1.0.0-xxx.el6.x86_64.rpm

- b) Unpack the disconnected installation package.

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

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


Insert the Red Hat/CentOS installation DVD.

The DVD contains some of the necessary software and utilities for installing TeamForge without internet access.

- d) Mount the Red Hat/CentOS 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 mount it as a filepath, with no spaces.

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

For example, if you are using vi:

```
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) Verify your yum configuration files.

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


Now, the package manager will look for the Red Hat installation components it needs 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 are 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 the 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/7.1.0.0
./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 7.1 as a fully-integrated add-on. TeamForge 7.1 supports Review Board 1.7.17.

For instructions on how to install Review Board, see TeamForge 7.1 [Administration Guide](#).

### Install Review Board

You must install the 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.

This procedure is for those who are installing the Review Board for the first time. In this scenario, both TeamForge and Review Board use PostgreSQL. TeamForge 7.1 supports Review Board 1.7.17.

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

1. Download the `RBInstaller-1.4.0.0.22.zip` file from <http://collab.net/downloads/integrations#tab-1> and save it in the `/var/ops/` folder.

2. Unzip the `RBInstaller-1.4.0.0.22.zip` file.

```
cd /var/ops/
unzip RBInstaller-1.4.0.0.22.zip
```

3. Modify these values in the `install.conf` file to suit your installation environment.

```
sudo vi /var/ops/RBInstaller-1.4.0.0.22/installer/install.conf
```

Option	Description
<code>rb_dir=/u1/reviewboard</code>	The path of the directory where the Review Board files and libraries are installed.
<code>rb_data_dir=/opt/collabnet/reviewboard/data</code>	The path of the directory where Review Board's database file, review request files and attachments are stored.
<code>domain=&lt;domain name or host name&gt;</code>	The Review Board site information. For example, <code>cu064.cloud.maa.collab.net</code> .
<code>rb_database_password=&lt;reviewboard_db_password&gt;</code>	The Review Board database password.
<code>rb_database_type=postgresql</code>	The Review Board database type.
<code>rb_database_name=&lt;reviewboard_db_name&gt;</code>	The Review Board database name.
<code>rb_database_user=&lt;reviewboard_username&gt;</code>	The Review Board database user name.
<code>rb_database_host=&lt;reviewboard_db_hostname&gt;</code>	The Review Board database host name.
<code>rb_database_port=&lt;reviewboard_db_port&gt;</code>	The Review Board database port.
<code>ctf_base_url=https://myapp.collab.net</code>	The absolute URL of the TeamForge site that you want to associate to.
<code>ctf_site_var_dir=/opt/collabnet/teamforge/var</code>	The location of the <code>rbctfevents.jar</code> 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 `install.py` script available in the Review Board installer directory.

```
cd /var/ops/RBInstaller-1.4.0.0.22
sudo python ./install.py -i -r --auth-scmuser
```

8. If this is an "advanced" mode installation (with the database on the same box or another box), do this on the database server.

- a) Stop the TeamForge application.

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

- b) 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/9.2/data/pg_hba.conf
host ctfdb ctfdbuser <IP address of my.app.box>/32 md5
$ exit
```

- c) Restart PostgreSQL.

```
sudo /etc/init.d/postgresql-9.2 restart
```

- d) Create the Review Board database and username.

```
$ su - postgres
$ createuser -P -S --no-createrole ctfdbuser
$ createdb -E UTF8 -O ctfdbuser ctfdb
$ exit
```

- e) Restart PostgreSQL.

```
sudo /etc/init.d/postgresql-9.2 restart
```

9. Set up the initial Review Board data. Run the `bootstrap-data.py` script available in the Review Board installer directory.

```
cd /var/ops/RBInstaller-1.4.0.0.22
sudo python ./bootstrap-data.py
```

10. Run the `grep` command to get the value of `httpd_user` and `httpd_group`.

```
grep httpd_user /etc/reviewboard.properties
grep httpd_group /etc/reviewboard.properties
sudo chown -R <httpd_user>:<httpd_group> /opt/collabnet/reviewboard/data
```


11. Restart the TeamForge application.

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

12. 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>
```

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 314 for more information.

-  **Note:** Run the following scripts available in the Review Board installer directory.

```
cd /var/ops/RBInstaller-1.4.0.0.22
```

- 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 416.

- 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 the 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.

This procedure is for those who are installing the Review Board for the first time on TeamForge sites that use Oracle database. While TeamForge uses Oracle, the Review Board uses PostgreSQL. TeamForge 7.1 supports Review Board 1.7.17.

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

1. Install PostgreSQL.

```
sudo yum install postgresql92-server postgresql92-libs
```

2. Download the RBInstaller-1.4.0.0.22.zip file from <http://collab.net/downloads/integrations#tab-1> and save it in the `/var/ops/` folder.

3. Unzip the RBInstaller-1.4.0.0.22.zip file.

```
cd /var/ops/
unzip RBInstaller-1.4.0.0.22.zip
```

4. Modify these values in the `install.conf` file to suit your installation environment.

```
sudo vi /var/ops/RBInstaller-1.4.0.0.22/installer/install.conf
```

Option	Description
<code>rb_dir=/u1/reviewboard</code>	The path of the directory where the Review Board files and libraries are installed.
<code>rb_data_dir=/opt/collabnet/reviewboard/data</code>	The path of the directory where Review Board's database file, review request files and attachments are stored.
<code>domain=&lt;domain name or host name&gt;</code>	The Review Board site information. For example, <code>cu064.cloud.maa.collab.net</code> .
<code>rb_database_type=postgresql</code>	The Review Board database type.
<code>rb_database_name=&lt;reviewboard_db_name&gt;</code>	The database name of the Review Board.
<code>rb_database_user=&lt;reviewboard_username&gt;</code>	The Review Board database user name.
<code>rb_database_password=&lt;reviewboard_db_password&gt;</code>	The Review Board database password.
<code>rb_database_host=&lt;reviewboard_db_hostname&gt;</code>	The Review Board database host name.
<code>rb_database_port=&lt;reviewboard_db_port&gt;</code>	The Review Board database port.
<code>ctf_base_url=https://myapp.collab.net</code>	The absolute URL of the TeamForge site that you want to associate to.
<code>ctf_site_var_dir=/opt/collabnet/teamforge/var</code>	The location of the <code>rbctfevents.jar</code> 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 `install.py` script available in the Review Board installer directory.

```
cd /var/ops/RBInstaller-1.4.0.0.22
sudo python ./install.py -i -r --auth-scmuser
```

9. Initialize the PostgreSQL database.

```
$ su - postgres
/usr/pgsql-9.2/bin/initdb -D /var/lib/pgsql/9.2/data
```

10. Edit the `postgresql.conf` file as a postgres user and update the following entry.

```
vi /var/lib/pgsql/9.2/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/9.2/data/pg_hba.conf
host ctfrbdb ctfrbuser <IP address of my.app.box>/32 md5
$ exit
```

12. Start the PostgreSQL service.

```
sudo /etc/init.d/postgresql-9.2 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-9.2 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-1.4.0.0.22
sudo python ./bootstrap-data.py
```

16. Run the `grep` command to get the value of `httpd_user` and `httpd_group`:

```
grep httpd_user /etc/reviewboard.properties
grep httpd_group /etc/reviewboard.properties
sudo chown -R <httpd_user>:<httpd_group> /opt/collabnet/reviewboard/data
```


17. Restart the TeamForge application.

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

18. 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>
```

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 314 for more information.

-  **Note:** Run the following scripts available in the Review Board installer directory.

```
cd /var/ops/RBInstaller-1.4.0.0.22
```

- 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 416.
- To uninstall Review Board, run the `sudo python ./install.py -u` command.

### Upgrade Review Board

Use these instructions to upgrade Review Board to a latest build.

This procedure is for those who have Review Board already and are upgrading Review Board to a latest build. You may choose to upgrade Review Board on the same server or on a new server. In this scenario, both TeamForge and Review Board use PostgreSQL. TeamForge 7.1 supports Review Board 1.7.17.

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

1. Download the `RBInstaller-1.4.0.0.22.zip` file from <http://collab.net/downloads/integrations#tab-1> and save it in the `/var/ops/` folder.
2. Uninstall the existing Review Board application if you are upgrading on the same server.

```
cd /var/ops/RBInstaller-1.3.0.0.22
sudo python ./install.py -u
```

3. Back up your Review Board data directory if you are upgrading TeamForge and Review Board on a new server. The Review Board database is backed up already when you have upgraded TeamForge. So, it is not necessary to take a back up of the Review Board database again.
  - a) Back up the Review Board data directory.

```
cd /opt/collabnet
tar -zcvf /tmp/reviewboard_data.tgz reviewboard
```

- b) Copy the `/tmp/reviewboard_data.tgz` file to the `/tmp` directory of the new server.

4. Unzip the `RBInstaller-1.4.0.0.22.zip` file.

```
cd /var/ops/
sudo unzip RBInstaller-1.4.0.0.22.zip
```

5. Modify these values in the `install.conf` file to suit your installation environment.

```
sudo vi /var/ops/RBInstaller-1.4.0.0.22/installer/install.conf
```

#### Option

**rb\_dir=/u1/reviewboard**

#### Description

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.

Option	Description
<code>domain=&lt;domain name or host name&gt;</code>	The Review Board site information. For example, <code>cu064.cloud.maa.collab.net</code> .
<code>rb_database_password=&lt;reviewboard_db_password&gt;</code>	The Review Board database password.
<code>rb_database_type=postgresql</code>	The Review Board database type.
<code>rb_database_name=&lt;reviewboard_db_name&gt;</code>	The database name of the Review Board.
<code>rb_database_user=&lt;reviewboard_username&gt;</code>	The Review Board database user name.
<code>rb_database_host=&lt;reviewboard_db_hostname&gt;</code>	The Review Board database host name.
<code>rb_database_port=&lt;reviewboard_db_port&gt;</code>	The Review Board database port.
<code>ctf_base_url=https://myapp.collab.net</code>	The absolute URL of the TeamForge site that you want to associate to.
<code>ctf_site_var_dir=/opt/collabnet/teamforge/var</code>	The location of the <code>rbctfevents.jar</code> 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 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>'
```

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 Review Board 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 to the `/tmp` directory of the new server.

- a) Restore the Review Board data directory.

```
cd /opt/collabnet
tar -zxvf /tmp/reviewboard_data.tgz
```

10. Run the `install.py` script available in the Review Board installer directory.

```
cd /var/ops/RBInstaller-1.4.0.0.22
sudo python ./install.py -i -r --auth-scmuser
```

11. Run the `migrate.py` script available in the Review Board installer directory.

```
cd /var/ops/RBInstaller-1.4.0.0.22
sudo python ./migrate.py
```

12. Restart the TeamForge application.

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

13. 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>
```

### Upgrade Review Board 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 the database.

This procedure is for those who have Review Board already and are upgrading Review Board to a latest build on TeamForge sites that use Oracle as the database. While TeamForge uses Oracle, the Review Board uses PostgreSQL. You may choose to upgrade Review Board on the same server or on a new server. TeamForge 7.1 supports Review Board 1.7.17.

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

1. Uninstall the existing Review Board application if you are upgrading on the same server.

```
cd /var/ops/RBInstaller-1.3.0.0.22
sudo python ./install.py -u
```

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> ctfrrdb -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. Download the RBInstaller-1.4.0.0.22 file from <http://collab.net/downloads/integrations#tab-1> and save it in the /var/ops/ folder.

4. Unzip the RBInstaller-1.4.0.0.22.zip file.

```
cd /var/ops/
sudo unzip RBInstaller-1.4.0.0.22.zip
```

5. Modify these values in the `install.conf` file to suit your installation environment.

```
sudo vi /var/ops/RBInstaller-1.4.0.0.22/installer/install.conf
```

Option	Description
<b>rb_dir=/u1/reviewboard</b>	The path of the directory where the Review Board files and libraries are installed.
<b>rb_data_dir=/opt/collabnet/reviewboard/data</b>	The path of the directory where Review Board's database file, review request files and attachments are stored.
<b>domain= &lt;domain name or host name&gt;</b>	Review Board site information. For example, cu064.cloud.maa.collab.net.
<b>rb_database_password=&lt;reviewboard_db_password&gt;</b>	The Review Board database password.
<b>rb_database_type=postgresql</b>	The Review Board database type.



Option	Description
<b>rb_database_name=ctfrbdb</b>	The Review Board database name.
<b>rb_database_user=ctfrbuser</b>	The Review Board database user name.
<b>rb_database_host=&lt;database host name&gt;</b>	The Review Board database host name.
<b>rb_database_port=&lt;reviewboard_db_port&gt;</b>	The Review Board database port.
<b>ctf_base_url=https://myapp.collab.net</b>	The absolute URL of the TeamForge site that you want to associate to.
<b>ctf_site_var_dir=/opt/collabnet/teamforge/var</b>	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) 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>'
```

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. Do this if you are upgrading TeamForge and Review Board on a new server.

- a) Stop the TeamForge application.

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

- b) Edit the pg\_hba.conf as a postgres user and add the following entry at end of the file.

```
su - postgres
/usr/pgsql-9.2/bin/initdb -D /var/lib/pgsql/9.2/data
vi /var/lib/pgsql/9.2/data/postgresql.conf
listen_addresses = '127.0.0.1,<IP address of app_box>'
exit
```

- c) Start the PostgreSQL service.

```
sudo /etc/init.d/postgresql-9.2 start
```


- d) Create the Review Board database and user name.

```
su - postgres
createuser -P -S --createdb --no-createrole ctfrbuser
createdb -E UTF8 -O ctfrbuser ctfrbdb
exit
```

- e) Restart PostgreSQL.

```
sudo /etc/init.d/postgresql-9.2 restart
```

10. Restore the Review Board data directory and database if you are upgrading TeamForge and Review Board on a new server.

 **Note:** Ensure that the Review Board data directory and database you backed up earlier has been copied to the /tmp directory of the new 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
```

11. Run the `install.py` script available in the Review Board installer directory.

```
cd /var/ops/RBInstaller-1.4.0.0.22
sudo python ./install.py -i -r --auth-scmuser
```

12. Run the `migrate.py` script available in the Review Board installer directory.

```
cd /var/ops/RBInstaller-1.4.0.0.22
sudo python ./migrate.py
```

13. Restart the TeamForge application.

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


14. 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>
```

### 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 305.

 **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: FAQs**

Frequently asked questions about TeamForge - Review Board integration.

#### **TeamForge - Review Board integration: Install FAQs**

Questions about TeamForge - Review Board integration.

**What are the software requirements for installing Review Board as an integrated application in TeamForge 7.1?**

Review Board can run on:

- RHEL 6.4
- CentOS 6.4
- SUSE Linux 11 SP2

In addition, Review Board needs PostgreSQL 9.2.4. See [Software requirements for CollabNet TeamForge 7.1](#) on page 356 topic for more information.

**Can I install TeamForge and Review Board on separate servers?**

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 7.1 support?**

TeamForge 7.1 supports Review Board 1.7.17.

**Which repositories does Review Board support?**

Review Board supports only Subversion repository in TeamForge 7.1.

**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 347

**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 Git

TeamForge 7.1 supports two versions of the Git integration — one based on Gerrit 2.1.10 (version 7.1.x of the integration) and the other based on Gerrit 2.8.x (version 8.2.x of the integration)

### Set up the TeamForge Git integration

The TeamForge Git integration is supported on RedHat and Centos version 5.6 or later. With TeamForge 7.1 (and later), you can install the Git integration as a part of installing TeamForge.

You can install Git integration on a separate box and other services on the main application box. For more information, see [Install TeamForge 7.1 with SCM and Git integration on a separate server](#) on page 34

You can upgrade to TeamForge 7.1 with the Git integration on the separate server. See [Upgrade to TeamForge 7.1 - GIT on a separate server](#) on page 147

### Set up Code Search for the TeamForge Git integration

To use TeamForge Code Search functionality for Git, manually grant the TeamForge Code Search user permissions to access all Git repositories.

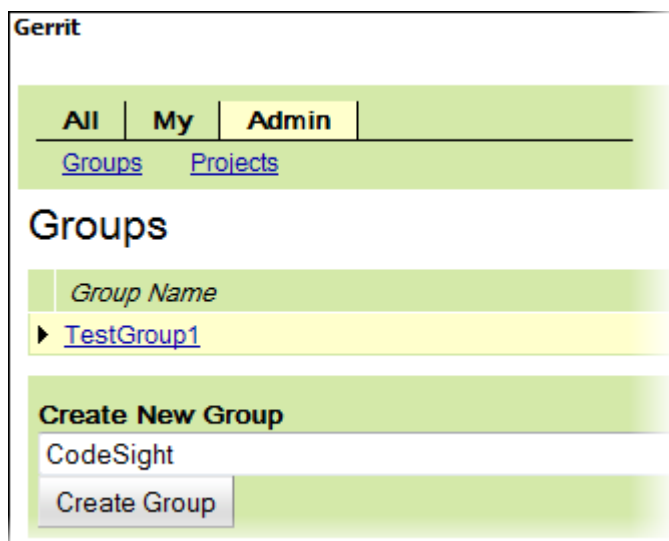
1. You need the root user's public key for SSH authentication on the Code Search server. Do the following:
  - a) On the Code Search box, check whether the key is present at `/root/.ssh/id_rsa.pub`. If not, generate it by running the `ssh-keygen` command.
  - b) Copy it to a temporary location (`/tmp`) on the TeamForge application server.
  - c) Run the `set_auth_key.py` script for the `scmviewer` user on the TeamForge application server.

```
cd /opt/collabnet/teamforge/runtime/scripts/codesearch/
./set_auth_key.py --authkey-file=/tmp/id_rsa.pub
```

2. Sync the `scmviewer` user to Gerrit by running the following command in a shell on the host where you installed the Git integration.

```
curl http://localhost:9081/api/gerrit/users/scmviewer/sshkeys
```

3. Log into the Gerrit console as a Gerrit super user and create an internal Gerrit group, for example, "CodeSight Group".



4. Add the `scmviewer` user to the group.
5. Grant read access to the group.
  - a) In the Gerrit project page that displays access rights, select **Read Access** for Category.

- b) For **Group Name**, enter the name of the internal group ("CodeSight" in the example) you created.
  - c) Enter "refs/\*" for **Reference Name**.
  - d) Enter "+1:Read Access" for **Permitted Rage**.
  - e) Click **Add Access Right**.
6. Log out from Gerrit.
  7. Restart the Code Search server.

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

## 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.

# 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/upgrade 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 7.1](#) on page 356.

## TeamForge 7.1 installation/upgrade FAQs

This section lists TeamForge 7.1 release-specific FAQs.

### Upgrade TeamForge 7.1 search index to Lucene 4.x format

TeamForge 7.1 and later uses Lucene 4.4. You can convert older search indices to Lucene 4.x format using the runtime script, `indexupgrade.py`, which is available with TeamForge 7.1 and later versions.

### Why should I run this `indexupgrade.py` script? What are the benefits?

The TeamForge 7.1 and later uses Lucene 4.4. As the index format of Lucene 4.4 differs from that of Lucene 3.x, post upgrade to TeamForge 7.1 or later, you may have to re-index your site or upgrade your existing indices to Lucene 4.4 format.

The search index files are typically huge in size (several Giga Bytes). If you choose to re-index data, it takes a lot of time and the search service would have to be down till then.

Instead, if you choose to upgrade your existing indices, you can convert your site's search indices to Lucene 4.x format quickly using the `indexupgrade.py` script with less downtime of the search service.

Converting the search indices to Lucene 4.x format would also improve the indexing and search performance considerably.

### Who should run this script and when?

Running this script is a one-time task. Customers upgrading from TeamForge 6.1.1, 6.2 (including Patch 1), or 7.0 to TeamForge 7.1 or later should run this script. No need to run this script if you are upgrading from TeamForge 7.0 to 7.1.

You may choose to run this script as part of the upgrade process (as recommended in the upgrade instructions, all the CollabNet services would be down as part of TeamForge upgrade and so would be the search service too) or at a later point in time post upgrade. However, with the latter case, you must bring the search service down prior to running the `indexupgrade.py` script.

### Should I back up the existing search index directory?

**Yes.** You must back up existing search index directory before running this script.

The `indexupgrade.py` script converts the existing search index directory to Lucene 4.x format. So, it is highly recommended that you back up the existing search index directory. Refer to the `SEARCH_INDEX_LOCATION` token in the `runtime-options.conf` file to know the search index directory's location.

### What is the recommended JVM heap size for `indexupgrade.py`?

Typically, a JVM heap size equivalent to the current index size is required. In the worst case, make sure you have a JVM heap size of at least half the size of the current index. For example, if the index size is 10 GB, the JVM heap size for `indexupgrade.py` should be at least 5 GB or more.

### Why do I get an exception when recreating the runtime?

A JBoss exception was encountered during TeamForge installation (during "recreate runtime"). This could be an issue with the JBoss vault library and is found to occur inconsistently. If you encounter this exception during installation, run the TeamForge install command (recreate runtime) again.

## Users invoked via su command in TeamForge

TeamForge invokes the super user (su) command during service start or stop.

The following are the operating system users that are invoked via the su command.

Users	TeamForge component that invokes the user	Purpose
sf-admin	TeamForge integration server (Tomcat)	During service start (collabnet start tomcat)
sf-admin	TeamForge ETL server (Tomcat)	During service start (collabnet start etl)
sf-admin	TeamForge application server (JBoss)	During service start (collabnet start jboss)
PostgreSQL	TeamForge database server (Postgres)	During service start/stop (postgresql-9.2 start/stop) and create runtime (install.sh)
bds-codesight	Code search database server (Postgres)	During service start/stop (bdc-codesight-postgresql start/stop) and create runtime (install.sh)
gerrit	Gerrit integration server (Jetty)	During service start (collabnet start gerrit)

## 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 272 for information.
- If your site requires SELinux, you must configure your Apache service. See [Set up SELINUX](#) on page 271 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.

<b>CollabNet TeamForge core functionality</b>	This is known internally as the <code>app</code> server. It implements JBoss and Apache services. One and only one instance of this application must be present.
<b>Database</b>	The <code>database</code> application handles site users' data. You set the type of database by setting the value of the <code>DATABASE_TYPE</code> token to <code>oracle</code> or <code>pgsql</code> . One and only one instance of this application must be present.
<b>Subversion</b>	Subversion can be used to provide source control functionality. It uses the Tomcat and Apache services. A site can have zero, one, or more than one instances of the <code>svn</code> application, running on an arbitrary number of machines.
<b>CVS</b>	CVS can be used to provide source control functionality. It uses the Tomcat service. A site can have zero, one, or more than one instances of the <code>cv</code> s application, running on an arbitrary number of machines.
<b>etl</b>	The reporting service.
<b>datamart</b>	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.

## 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.

### Recommended values if PostgreSQL and TeamForge are on the same server

site-options.conf tokens	8 GB RAM	16 GB RAM	32 GB RAM	64 GB RAM	128 GB RAM
PGSQL_EFFECTIVE_CACHE_SIZE	6 GB	12 GB	24 GB	48 GB	96 GB
PGSQL_SHARED_BUFFERS	2 GB	4 GB	8 GB	8 GB	8 GB
PGSQL_WORK_MEM	8 MB	16 MB	32 MB	64 MB	64 MB
PGSQL_WAL_BUFFERS	32 MB	32 MB	32 MB	32 MB	32 MB
PGSQL_MAINTENANCE_WORK_MEM	615 MB	615 MB	615 MB	615 MB	615 MB

### Recommended values if PostgreSQL is on a separate server

site-options.conf tokens	8 GB RAM	16 GB RAM	32 GB RAM	64 GB RAM	128 GB RAM
PGSQL_EFFECTIVE_CACHE_SIZE	12 GB	24 GB	48 GB	96 GB	96 GB
PGSQL_SHARED_BUFFERS	4 GB	8 GB	8 GB	8 GB	8 GB
PGSQL_WORK_MEM	16 MB	32 MB	64 MB	64 MB	64 MB
PGSQL_WAL_BUFFERS	32 MB	32 MB	32 MB	32 MB	32 MB
PGSQL_MAINTENANCE_WORK_MEM	615 MB	615 MB	615 MB	615 MB	615 MB

## 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](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 7.1 on a new box?

You can upgrade TeamForge on new hardware, or you can stick with the same box your current site is running on. To choose, consider how your members use your site and how you maintain it.

### Upgrade on new hardware

Upgrading on new hardware is a little more complex, but provides more flexibility.

Do it this way if you want to:

- Change to a different operating system.
- Stage your upgraded site before moving it into production.
- Serve the site from a different data center.
- Support an expanded user base.
- Minimize downtime associated with the upgrade.

### Upgrade on the same hardware


If you are OK with your current hardware setup and you don't want to rock the boat, you can run your new TeamForge site on the same box where your current site is running.

Do this if you want to:


- Keep your upgrade as simple as possible.
- Stick with a tested hardware setup.
- Use virtual hardware.
- Support a stable user base.

## Should I move my TeamForge database to its own server?

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

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

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

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

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

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

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

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

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

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

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

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

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

## 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](http://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 "-nogpgcheck 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 7.1. 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 7.1. 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 7.1 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](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 of 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.
```

## 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 TeamForge to 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 pgsq1
```

**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/runtime/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 272 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 7.1

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
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 `/cvssroot` 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 `cvsserver` (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 authentication, 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 algorithm<sup>1</sup>. 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 in 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 `<servername...></servername>` 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@appl 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
<b>Stale process issue</b>	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.



Issue	Stellent	Apache TIKA
<b>Search queue processing speed</b>	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.
<b>Multiple processes Vs Single JVM</b>	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 more resources, other processes and applications are left with scarce resources.	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.

## 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 vessages.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/vessages.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 YYYYMMDD>` 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.

<b>Blog Reader</b>	You can only read blogs and make comments, the comments are sent for moderation.
<b>Blog contributor</b>	You can add blog posts, but they will be sent for moderation.
<b>Blog publisher</b>	You can add blog posts, moderate comments and blog posts.
<b>Blog owner</b>	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 395 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 281 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 283.

## 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 Administrator 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 7.1

---

This is the hardware and software platform that TeamForge runs on.

#### Hardware requirements for CollabNet TeamForge 7.1

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, it is highly recommended that you add more RAM, which is to be exclusively allocated to PostgreSQL. CollabNet recommends adding 50% more RAM for TeamForge 7.1 compared to TeamForge 6.2. Contact CollabNet Support for tuning the application for the additional RAM.

#### Software requirements for CollabNet TeamForge 7.1

This is the official list of software tested for compatibility with CollabNet TeamForge 7.1.

##### Software version requirements for 32/64 bit platform

	Red Hat Enterprise Linux 6.4 <sup>#</sup>	CentOS 6.4	SUSE Linux Enterprise Server 11 SP2 <sup>#</sup>
Databases PostgreSQL <sup>#</sup>	9.2.4	9.2.4	9.2.4
Oracle DB <sup>#</sup>	11G (R1 and R2)	11G (R1 and R2)	11G (R1 and R2)

<sup>#</sup> PGTurant tool is available for quick pg\_upgrade. See [Upgrade PostgreSQL using PGTurant](#) on page 256 for more information.

	Red Hat Enterprise Linux 6.4 <sup>#</sup>	CentOS 6.4	SUSE Linux Enterprise Server 11 SP2 <sup>#</sup>
Oracle Client <sup>#</sup>	11g	11g	11g
Java <sup>#</sup> JDK	1.7.0_40	1.7.0_40	1.7.0_40
Jboss	7.1.1	7.1.1	7.1.1
Tomcat	7.0.22	7.0.22	7.0.22
Lucene	4.4	4.4	4.4
Open LDAP	2.4.23	2.4.23	2.4.26
Quartz Job Scheduler	1.8.3	1.8.3	1.8.3
BrowserGoogle Chrome	30	30	30
Mozilla Firefox	25	25	25
Microsoft IE	8, 9, 10 and 11 <sup>#</sup>	8, 9, 10 and 11 <sup>#</sup>	8, 9, 10 and 11 <sup>#</sup>
yum package manager	3.2.29 or earlier versions	3.2.29 or earlier versions	-
zypper package manager	-	-	1.6.161 or earlier versions
<b>CollabNet TeamForge 7.1 supports the following software versions for integration</b>			
Black Duck Code Sight <sup>#</sup>	2.1.3	2.1.3	2.1.3
Review Board <sup>#</sup>	1.7.17	1.7.17	1.7.17
GIT	1.7 or later	1.7 or later	1.7 or later
Subversion	1.8.3	1.8.3	1.8.3
Subversion Edge	4.0.3	4.0.3	4.0.3
CVS	1.11.x	1.11.x	1.11.x
Perforce	2012	2012	2012
ViewVC	1.1.20	1.1.20	1.1.20
<ul style="list-style-type: none"> <li>• <sup>#</sup> Red Hat Enterprise Linux servers must have access to the <a href="#">Red Hat Network</a> or equivalent (satellite server, spacewalk, or RHN proxy).</li> <li>• <sup>#</sup> SUSE Linux Enterprise Server must be registered with Novell.</li> <li>• <sup>#</sup> Oracle Express Edition is not supported.</li> <li>• <sup>#</sup> For larger installs, the max JVM heap size should be increased to 1024MB (or larger), depending on available resources on the server.</li> <li>• <sup>#</sup> Black Duck Code Sight 2.1.3 tested with PostgreSQL9.2.4 only. Black Duck Code Sight with Oracle was not tested.</li> <li>• <sup>#</sup> Review Board 1.7.17 tested with PostgreSQL 9.2.4 only. Review Board with Oracle was not tested.</li> <li>• <sup>#</sup> TeamForge 7.1 has been tested for compatibility with Internet Explorer 11 (patch number: 11.0.9600.17239).</li> </ul>			

## CollabNet TeamForge 7.1 on VMware and ESXi

The CollabNet TeamForge 7.1 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 7.1 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 - TeamForge 7.1

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)	2.2.15-29
httpd (CentOS)	2.2.15-29
jdk	1.7.0_40
mod_dav_svn	1.8.3-1
mod_python	3.3.1-15
mod_ssl (Red Hat)	2.2.15-29
mod_ssl (CentOS)	2.2.15-29
neon	0.29.3-2 or later
serf	1.2.1-3
oracle-instantclient11.2-basic	11.2.0.1.0-1
postgresql	9.2.4
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 - 1.8.3-1 subversion-python - 1.8.3-1

Package	Version Number
	subversion-perl - 1.8.3-1 subversion-devel - 1.8.3-1
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 - TeamForge 7.1

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_40
oracle-instantclient11.2-basic	11.2.0
postgresql	9.2.4
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 - 1.8.3-1 subversion-python - 1.8.3-1 subversion-perl - 1.8.3-1 subversion-devel - 1.8.3-1
ZSI	2.0-6 or later.

## Hardware and software requirements for CollabNet TeamForge 7.1 on a virtual machine

This is the hardware and software platform requirements if you run TeamForge on VMware Player.

- An operating system that VMware Player can run on. For Windows and Linux versions supported by VMware Player, see the [VMware Player documentation](#).
- 4 GB system RAM
- 8 GB available disk space
- 2 Ghz Pentium 4 or equivalent processor

 **Note:** CollabNet TeamForge may also run on some other VMware products, such as VMware Workstation 5.5. However, these instructions are only for running TeamForge on VMware Player.

## Scripts installed with TeamForge

---

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>-b   --both</b>	Back up both the database and the filesystem. This is the default option.
<b>-d   --database</b>	Back up the database.
<b>-f   --files</b>	Back up the filesystem.
<b>-h   --help</b>	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:

<b>-d   --directory</b>	Specify installation directory. This argument is required.
<b>-h   --help</b>	Provides a list of all available options for this script.
<b>-n   --non-interactive</b>	Runs the script in a non-interactive mode. The script will fail with an error message when used with this option if an existing <code>[DATA_DIR]</code> is located. You can use the <code>-F</code> option to force bootstrapping on sites that have an existing <code>[DATA_DIR]</code> .
<b>-F   --force</b>	This option is only valid when the <code>-n</code> option is used. This option forces the bootstrapping of data when a <code>[DATA_DIR]</code> exists.
<b>-V   --verbose</b>	Writes all script actions to the screen. Without this option the script runs silently and logs messages to the <code>[log_file_directory]/runtime/bootstrap.log</code> file.
<b>-q   --quiet</b>	Do not show script output.
<b>-f   --site-options=[filename]</b>	Points to the <code>site-options.conf</code> configuration file for the site. This argument is optional.


## Cluster location

This script runs on the application server machine.

## 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:

<b>-h   --help</b>	Provides a list of all available options for this script.
<b>-n   --non-interactive</b>	Runs the script in a non-interactive mode. The script will fail with an error message when used with this option if an existing <code>[REPORTS_DATA_DIR]</code> is located. You can use the <code>-F</code> option to force bootstrapping on sites that have an existing <code>[REPORTS_DATA_DIR]</code> .
<b>-F   --force</b>	This option is only valid when the <code>-n</code> option is used. This option forces the bootstrapping of data when a <code>[DATA_DIR]</code> exists.
<b>-V   --verbose</b>	Writes all script actions to the screen. Without this option the script runs silently and logs messages to the <code>[log_file_directory]/runtime/bootstrap.log</code> file.
<b>-q   --quiet</b>	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:

<b>--target=create</b>	This is the default target and is used for creating Black Duck Code Sight projects for existing TeamForge repositories.
<b>[--filename=&lt;repositories-input-filename&gt;]</b>	Default: <code>/tmp/repository-ids</code> ; specify any other

		filename with the --filename parameter.
	<b>--username=&lt;admin-username&gt;</b>	Site admin username
	<b>--password=&lt;admin-password&gt;</b>	Site admin password
	<b>[--totalthreads=&lt;number of threads to spawn&gt;]</b>	Default: 25; the total number of threads needed to run Black Duck Code Sight migration.
<b>--target=delete</b>	Delete Black Duck Code Sight projects for the corresponding TeamForge repositories.	
	<b>[--filename=&lt;repositories-input-filename&gt;]</b>	Default: /tmp/repository-ids; specify any other filename with the --filename parameter.
	<b>--username=&lt;admin-username&gt;</b>	Site admin username
	<b>--password=&lt;admin-password&gt;</b>	Site admin password
	<b>[--repoids=&lt;repo-id1,repo-id2,repo-id3&gt;]</b>	Default: delete all repositories as specified in the input filename or delete the repositories specified in the command line
<b>--target=status</b>	Status of Black Duck Code Sight projects created for TeamForge repositories.	
	<b>[--filename=&lt;repositories-input-filename&gt;]</b>	Default: /tmp/repository-ids; specify any other filename with the --filename parameter.
	<b>--username=&lt;admin-username&gt;</b>	Site admin username
	<b>--password=&lt;admin-password&gt;</b>	Site admin password
	<b>[--repoids=&lt;repo-id1,repo-id2,repo-id3&gt;]</b>	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:
	<p><b>start</b> Starts the application / service</p> <p><b>stop</b> Stops the application / service</p> <p><b>status</b> Provides the status of the service(s)</p> <p><b>restart</b> Restarts the application / service</p>
<b>help</b>	Prints this message and exits.
<b>-s   service serviceName</b>	Perform the command for the service serviceName
<b>verbose</b>	Print debug messages
<b>-q   --quiet</b>	
<b>-F   --force</b>	Force option to perform the specified operation forcefully
<b>-S   --silent</b>	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:

<b>-t</b>   <b>--type</b>	Specifies the type of database (ctf or reporting).
<b>-h</b>   <b>--help</b>	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:


<b>--help</b>	Show this message
<b>--debug</b>	Include debugging output
<b>--old</b>	Old domain
<b>--new</b>	New domain
<b>--dir</b>	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 <code>domain_name</code> and replacing it with the new <code>domain_name</code> .
	 <b>Note:</b> Without this option, only HTML, text, and VM files are modified.
<b>--threadlimit</b>	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:

<b>--help</b>	Show this message
<b>--debug</b>	Include debugging output
<b>--old</b>	Old domain
<b>--new</b>	New domain
<b>--dir</b>	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 <code>domain_name</code> and replacing it with the new <code>domain_name</code> .
	 <b>Note:</b> Without this option, only HTML, text, and VM files are modified.
<b>--threadlimit</b>	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:

<b>-s   --status</b>	Prints the status of all the jobs configured in the ETL service.
<b>-a   --status-all</b>	Prints the status of incremental and historical jobs configured in the ETL service.
<b>-v   --verbose</b>	Chronicles the process of requested operation a bit more.
<b>-r   --run=</b>	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
<b>Historical Data Collection</b> 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.
<b>Incremental Data Collection</b> 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.
<b>Imported Data Collection (Simbel)</b> 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:

<b>-r</b>	Number of repos to process at a time. Default=5
<b>-d</b>	Number of directories to process at a time. Default=<No of processors>
<b>-t</b>	Item to fix. Possible values are [app, http, svn, cvs, git]. Default=<all items>
<b>-h   --help</b>	Prints the help information.

## install.sh

The `install.sh` script handles all operations related to installing and removing the application.

### Options

<b>-a   --all</b>	Performs these operations in sequence - Install, create runtime and setup initscripts (equivalent to: -I -r -i)
<b>-b   --bootstrap</b>	Sets up initial site data.
<b>-C   --cleanup</b>	Stop/Kill the application processes, wipes out application packages and site directory.
 <b>Caution:</b>	THIS OPTION WILL WIPE OUT THE SITE DATA.
<b>-c   --conf-file</b>	The environment configuration file (usually of the form environment-<platform>.conf), used to identify the platform.
 <b>Note:</b>	This is not to be confused with the the <code>site-options.conf</code> file.
<b>-d   --directory type='str', argname='installation_base_dir',</b>	Path where the site would get installed.
<b>-E   --check_environment</b>	Check if the system environment is suitable for this installation.
<b>-F   --force</b>	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**

<b>--help   -h</b>	Print out a help message and exit.
<b>--api-user   -u username</b>	Your TeamForge Lab Management login name. Required for all upload operations.
<b>--api-key   -k key</b>	Your TeamForge Lab Management API key. Required for all upload operations.
<b>--api-url   -l url</b>	The URL to the TeamForge Lab Management API's. Will generally be <code>https://\$external_host/cubit_api/1</code> . Required for all upload operations.
<b>--comment   -c "your comment"</b>	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.
<b>--verbose   -v</b>	Print out more detail on what the <code>pbl.py</code> is doing.
<b>--xml-server-output   --xml</b>	If this option is not specified, the <code>pbl.py</code> client reads in the XML returned from the server and presents the results to you in nicely formatted text. If you'd like to

<code>--no-auth-cache</code>	instead see the raw XML returned from the server, select this option.  As a convenience, the <code>pbl.py</code> function caches the value of the <code>--api-user</code> and <code>--api-key</code> parameters in your home directory, in a subdirectory named <code>.cubit</code> , the first time a successful authentication is performed against the server. This is analogous to the Subversion client's use of the <code>.subversion</code> directory to store authentication credentials. Selecting the <code>--no-auth-cache</code> option turns off this caching.
<code>--project p projname</code>	The TeamForge Lab Management project in which the file you are operating on is located.
<code>--type t {pub priv}</code>	The visibility type of the file, either <code>pub</code> (the file is in the public area of the PBL) or <code>priv</code> (the file is in the private area of the PBL).
<code>--remotepath r path</code>	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>-b   --both</b>	Restore both the database and the filesystem. This is the default option.
<b>-d   --database</b>	Restore the database.
<b>-f   --files</b>	Restore the filesystem.
<b>-h   --help</b>	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

<code>--single-item itemId,   -i</code>	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.
<code>---force-index   -f</code>	Force indexing (doesn't check if item is searchable already).
<code>--artifacts only   -a</code>	Reindex all artifacts on the site that are currently not searchable or all artifacts if option f is selected.
<code>--documents only   -d</code>	Reindex all documents on the site that are currently not searchable or all artifacts if option f is selected.
<code>---posts only</code>	Reindex all posts on the site that are currently not searchable or all artifacts if option f is selected.
<code>---trackers only</code>	Reindex all trackers on the site.
<code>---document_folders-only</code>	Reindex all document folders on the site.
<code>---topics-only</code>	Reindex all topics.
<code>---forums-only</code>	Reindex all forums on the site.
<code>---news-only</code>	Reindex all news.
<code>---project_pages-only</code>	Reindex all project pages.
<code>---packages</code>	Reindex all packages.

<code>---commits-only</code>	Reindex all commits.
<code>---frs_files-only</code>	Reindex all frs files.
<code>---releases-only</code>	Reindex all releases.
<code>---wikis-only</code>	Reindex all wikis.
<code>--project-id projectID   -p</code>	Limit the re-indexing to data for single project when re-indexing only artifacts and or documents.
<code>--verify   -x</code>	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.
<code>--dryrun   -r</code>	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.
<code>output-file filePath,   -o</code>	Prints the output for the given file.
<code>--verbose   -v</code>	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:

<b>-h --help</b>	Provides information on using the script.
<b>--extra</b>	An arbitrary command whose output should be placed in the generated log file. For example, you can have <code>snapshot.py</code> execute the <code>lsof</code> command like this: <pre>--extra '/usr/sbin/lsof -n -P -b -i -U'</pre> Enclose commands with options in quotes.
<b>-v --verbose</b>	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

<b>-f [manifest]   --file [manifest]</b>	The manifest file with the appropriate information for this upgrade.
<b>-d [INSTALLATION_DIR]   --directory [INSTALLATION_DIR]</b>	The directory where the application is installed.  <b>Note:</b> This option is required.
<b>-r   --rollback</b>	Rolls back the previous (most recently applied) patch. For example, if you upgrade the site from patch level 1 to patch level 4, and then run <code>upgrade-site.sh</code> with this option, the resulting patch level on the site is patch 1.
<b>-l [level]   --level [level]</b>	The patch level to which the SourceForge site must be upgraded (or downgraded).
<b>-V   --verbose</b>	Displays script output including traceback errors. If this option is not used, the script displays error messages but not the actual traceback errors.
<b>-v   --version</b>	Displays the script version.
<b>-n   --noninteractive</b>	Non-interactive mode.
<b>-t   --testrun</b>	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.  <b>Note:</b> You must use this option along with the <code>l</code> , <code>r</code> , <code>u</code> , or <code>f</code> options.
<b>-u   --uninstall</b>	Decrements the patch level on the site by one. For example, if you upgrade the site from patch level 1 to patch level 4, and then run <code>upgrade-site.sh</code> with this option, the resulting patch level on the site is patch 3.
<b>-h   --help</b>	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 `projecttracker.py` script.

 **Note:** "Catalina" is the code name for the integrated Project Tracker feature.

<b>debug -security</b>	Debug Catalina with a security manager
<b>jpda start</b>	Start Catalina under JPDA debugger
<b>run</b>	Start Catalina in the current window
<b>run -security</b>	Start in the current window with security manager
<b>start</b>	Start Catalina in a separate window
<b>start -security</b>	Start in a separate window with security manager
<b>stop</b>	Stop Catalina
<b>stop -force</b>	Stop Catalina (followed by kill -KILL)
<b>version</b>	What version of tomcat are you running?
<b>restart</b>	Stop and Start Catalina
<b>status</b>	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 WMTTool, click [https://forge.collab.net/sf/wiki/do/viewPage/projects.sf\\_engine/wiki/WMTTool](https://forge.collab.net/sf/wiki/do/viewPage/projects.sf_engine/wiki/WMTTool)

### 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:

<b>-a   -- All projects</b>	Runs the script for all the projects.
<b>-s   -- Specific project</b>	Runs the script for specific projects by mentioning the project name.

## TeamForge 7.1 scripts

This section lists TeamForge 7.1 release-specific scripts.

### indexupgrade.py

Run the `indexupgrade.py` script post upgrade to convert existing search indices to Lucene 4.x format.


### Overview


See [Upgrade TeamForge 7.1 search index to Lucene 4.x format](#) on page 318.



**Warning:** You must back up the existing search index directory before running this script.

### Usage

Command	Description
<code>sudo ./runtime/scripts/indexupgrade.py -h</code>	Displays the Help text for the script.
<code>sudo ./runtime/scripts/indexupgrade.py</code>	Upgrades the existing search indices to Lucene 4.x format.  <b>Important:</b> If the upgrade fails with an out of memory error (OOM), increase the maximum heap size for JVM and run the script again.  Typically, a JVM heap size equivalent to the current index directory's size is required. In the worst case, make sure you have a JVM heap size of at least half

Command	Description
	the size of the current index directory. For example, if the index size is 10 GB, the JVM heap size for <code>indexupgrade.py</code> should be at least 5 GB or more.
<pre>sudo ./runtime/scripts/indexupgrade.py --max-heap-size=N</pre> <p> <b>Note:</b> Where N is the maximum heap size for JVM.</p>	Upgrades the existing search indices to Lucene 4.x format. The maximum JVM heap size is passed as one of the parameters to provision additional memory for the JVM if the search index directory is considerably huge (typically the case with customers having huge amount of data in terms of several GB).

### Logging

The `indexupgrade.py` script writes entries to: `<SITE_DIR>/log/runtime/indexupgrade.log`.

## Log files in TeamForge - 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`

Logs the JBoss startup and shut down notifications. This log is overwritten each time JBoss is (re)started.

#### `localhost_access`

The Records access to the application from a remote host, similar to the Apache `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 - 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/7.1.0.0/conf`.)

When TeamForge starts, it reads and implements the instructions provided as values to the variables in this file.

The basic procedure for configuring your TeamForge site, then, is to edit the `site-options.conf` file, supply a valid value for the variable of interest, save the file, and restart the TeamForge runtime environment.

TeamForge has a number of configuration files that can be used to improve site performance. You can configure the `site-options-small.conf`, `site-options-medium.conf` and `site-options-large.conf` files to suit the needs of small, medium and large systems. Each of these sample configuration files has information about the load it supports and the hardware (CPU and memory) required.

**FILTER\_DROPDOWN\_MAX\_SELECTION**

By default, the drop-down lists with multi-select feature (available since TeamForge 7.1) let you select up to five filter values. However, you can set any value that suits your requirement for this *FILTER\_DROPDOWN\_MAX\_SELECTION* token to increase or decrease the count.

**Values**

Any positive integer.

**Default**

5

**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\_NO\_PASSWORD\_ON\_USER\_CREATION**

The *ALLOW\_NO\_PASSWORD\_ON\_USER\_CREATION*, when set to true, allows the admin to create users with null password through SOAP when external authentication is enabled.

**Values**

true, false

**Default**

false



**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**

Plain Text

**Default**

Plain Text

**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**

Plain Text

**Default**

Plain Text

**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**

Plain Text

**Default**

Plain Text

**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* token, if set with a preferred time zone, takes precedence over the physical TeamForge server location's time zone and will be the default time zone that's displayed throughout the application. In other words, use this token to set a preferred time zone to display across the TeamForge application in case the TeamForge physical server and users are not on the same time zone.

**Values**

The ID for a time zone 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**

If set, this token overrides the default time zone of the TeamForge server. TeamForge uses the default time zone of the JVM otherwise.



**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**

Plain Text

**Default**

Plain Text

**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\_apbbox.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.

**GERRIT\_GIT\_PUSH\_THRESHOLD**

The *GERRIT\_GIT\_PUSH\_THRESHOLD* token determines the maximum number of commits in a single Git push. If the limit exceeds, only a single commit object is created in the TeamForge.

**Values**

Any positive integer.

**Default**

30

**GERRIT\_GIT\_REFRESH\_PERIOD**

The *GERRIT\_GIT\_REFRESH\_PERIOD* token sets the interval in seconds after which Git Integration synchronizes all the repositories and all RBAC permission with TeamForge.

**Values**

Seconds.

**Default**

3600 Seconds.

**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**

An administrator can optionally supply a password when creating a user. If the password is not specified while creating the user, the user is sent an email with a ticket to set the password. This *INITIAL\_PASSWORD\_CHANGE\_ACTIVATION\_CODE\_TIMEOUT* token sets the duration (in hours) for which the password ticket is valid.

**Values**

Integer (hours)

**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 7.1 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

**LISTEN\_BACKLOG**

The *LISTEN\_BACKLOG* token is used to specify the maximum length of the queue for the pending connections in the Apache server.

**Values**

Integer

**Default**

The default value is obtained from the system Kernel configuration.

```
/sbin/sysctl -n -e net.ipv4.tcp_max_syn_backlog
```

**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 414 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.

**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.

See [What are the right PostgreSQL settings for my site?](#) on page 321 for values recommended by CollabNet.

**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.

**Comments**

See [What are the right PostgreSQL settings for my site?](#) on page 321 for values recommended by CollabNet.

**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.

**Comments**

The right value for this variable depends in part on the available RAM on the server where your site is running.

See [What are the right PostgreSQL settings for my site?](#) on page 321 for values recommended by CollabNet.

**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.

**Comments**

If your database has many write transactions, setting this value bit higher than default may result better usage of disk space.

See [What are the right PostgreSQL settings for my site?](#) on page 321 for values recommended by CollabNet.

**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. .

**Comments**

The right value for this variable depends in part on the available RAM on the server where your site is running.

See [What are the right PostgreSQL settings for my site?](#) on page 321 for values recommended by CollabNet.

**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\_DESC\_EDITOR**

The *PLANNING\_FOLDER\_DESC\_EDITOR* variable allows you to choose the type of text that can be used in the planning folder description using the editor tool.

**Values**

Plain Text

**Default**

Plain Text

**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 postgres 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_PASSWORD`, 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 406.

**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 371 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 certificate 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

**SVN\_AUTHNZ\_TIMEOUT**

The *SVN\_AUTHNZ\_TIMEOUT* token allows you to set the timeout value (in seconds) for the `mod_authnz_ctf` module.

**Values**

Timeout value in number of seconds.

**Default**

60 seconds

**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			

Name	Description	Default	Values	Comments
	example, /u1/ sourcecast.			
PROJECT_TRACKER_XML_PATH	XML file in the PT XML file, for example, /u3/ PTInstaller-1.0.45/ installer/ conf/pt.xml.			
PROJECT_CREATION_DATE	The 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>ll10n.application.description</description>
  <permissions>
    <permission dapMappedTo="View">Blog Reader</permission>
    <permission>Blog Contributor</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>
  </bundle>
</bundles>

```



```

        <key name="110n.three.value">3</key>
        <key name="110n.five.value">5</key>
        <key name="110n.seven.value">7</key>
        <key name="110n.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 validated. 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 as 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
```

## login-config.xml

This is the sample `application-policy` block that you can copy into your `login-config.xml` file to support LDAP authentication.

### Notes

Replace the default `application-policy` block of the `login-config.xml` file with this code, then make the modifications specified in [Set up LDAP integration for the CollabNet TeamForge site](#) on page 261. Option values that must be modified are highlighted in bold>.

- When the username is passed to the login module from TeamForge, it is translated into a DN for lookup on the LDAP server. The DN that is sent to the LDAP server is `<principalDNPrefix><username><principalDNSuffix>`.
- In this example `application-policy` block, the username is stored in the People organizational unit in the `dev.sf.net` domain. This is represented as `,ou=People,dc=dev,dc=sf,dc=net`
- This example contains a single `login-module` section. If you are authenticating against multiple LDAP servers, include one `login-module` section per LDAP server, with the required option values modified appropriately for each one. If the same username exists in more than one LDAP server, the instance on the first LDAP server will be used.

### Sample code

```
<application-policy name="SourceForge">
  <authentication>
    <login-module code="org.jboss.security.auth.spi.LdapLoginModule"
flag="sufficient" >
      <module-option name="allowEmptyPasswords">false</module-option>
      <module-option name="principalDNPrefix">uid=</module-option>
      <module-option
name="principalDNSuffix">,ou=People,dc=dev,dc=sf,dc=net</module-option>
```

```

        <module-option
name="java.naming.factory.initial">com.sun.jndi.ldap.LdapCtxFactory</module-
option>
        <module-option name="java.naming.provider.url">ldap://
util.dev.sf.net:389</module-option>
        <module-option
name="java.naming.security.authentication">simple</module-option>
    </login-module>
</authentication>
</application-policy>

```

### Sample code for Active Directory integration

Active Directory is not supported. However, these sample lines in the `login-config.xml` file may help you make it work for a simple AD setup, without complex directory structures requiring additional search parameters.

Set the values of `java.naming.provider.url`, `principalDNSuffix` and `rolesCtxDN` as appropriate to your site.

For more detailed instructions, see <http://www.jboss.org/community/wiki/LdapLoginModule>.

```

        <login-module code="org.jboss.security.auth.spi.LdapLoginModule"
flag="required" >
        <module-option name="java.naming.provider.url">ldaps://
<server_name>:636</module-option>
        <module-option name="allowEmptyPasswords">false</module-option>
        <module-option name="principalDNSuffix">@foo.bar.com</module-
option>
        <module-option name="rolesCtxDN">dc=Foo,dc=Bar,dc=Com</module-
option>
        <module-option name="matchOnUserDN">true</module-option>
        <module-option name="uidAttributeID">sAMAccountName</module-
option>
        <module-option name="roleAttributeID">memberOf</module-option>
        <module-option name="roleAttributeIsDN">true</module-option>
        <module-option name="roleNameAttributeID">name</module-option>
    </login-module>


```

## The patch manifest file

The patch manifest file contains all the information about the patch.

### Overview

The manifest file for each patch is named `manifest-[patch#]`. The manifest file is a text file containing a set of configuration tokens.


 **Note:** The first patch is named `manifest-1`.

### Contents

The manifest file contains these tokens:

**PATCH\_LEVEL**

The patch level which this patch provides.

 **Note:** The `PATCH_LEVEL` value is used (along with information in `[DISTRIBUTION_DIR]/version/`

core-version.txt ) to fill in  
 [DISTRIBUTION\_DIR]/conf/patches  
 with the current release and patch level. If  
 [DISTRIBUTION\_DIR]/conf/patches  
 does not exist, it is created.

**PATCH\_DESCRIPTION**


A description of the patch.

**UNINSTALL\_LIST**

A list of RPMs to uninstall (using relative paths, comma separated).

**INSTALL\_LIST**

A list of RPMs to install (using relative paths, comma separated).

 **Note:** Comments in the manifest file are identified by a leading hash (#).

## postgres.conf

The `/var/lib/pgsql/9.0/data/postgresql.conf` file controls the behavior of the PostgreSQL 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 7.1 release notes

---

Released on: 19 December, 2013

TeamForge 7.1 is here to make you more agile and accomplish more in less time and with fewer clicks.

## New features in CollabNet TeamForge 7.1

---

TeamForge 7.1 has a lot of new features such as improved password security configuration management, improved planning board, filter drop-down lists with multi-select capability, context menu for quickly creating dependencies, associations and adding attachments and more.

Here's a list of a few release-defining new features in TeamForge 7.1.

- *Enhanced password security configuration management*
- *Improved planning board*
- *Multi-select drop-down lists for filtering*
- *Context menu for quickly creating dependencies, associations and adding attachments*
- *Extended path-based permissions with partial wildcard match*
- *System tool to change a site's log level*
- *Responsive drop-down and scrollable menus for mobile devices*
- *Site-wide role based permission for administrating Review Board*
- *Change tracker type during artifact edits*
- *Highcharts-based interactive charts*
- *Multi-threaded search indexing*
- *Two new life cycle metric charts: Release Burn Up Chart and Release Burn Down Chart*
- *Upgrades to software infrastructure*

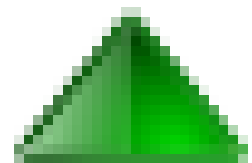
### Enhanced password security configuration management

#### Automatic password generation

TeamForge 7.1 installer supports automatic password creation for password-related `site-options.conf` tokens. By default, password-related tokens are preset with a value of `$auto$` in the `site-options.conf` file. As a result, the passwords for the tokens are randomly generated and stored in an encrypted format in the `site-options.conf` file (passwords are encrypted only if password obfuscation is enabled). This feature is enabled by default. You can, however, override any of the password-related tokens with the password of your choice.

#### Password obfuscation

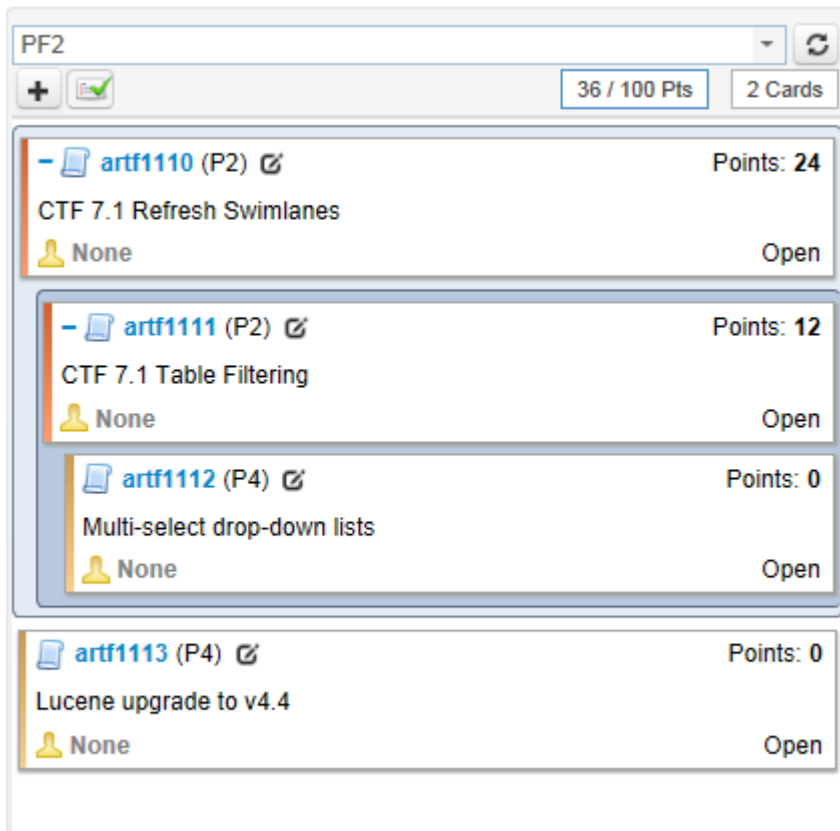
The password obfuscation is enabled by default (`OBFUSCATION_ENABLED=true` by default). As a result, all password-related tokens are encrypted in all the TeamForge configuration files. To disable password obfuscation, set `OBFUSCATION_ENABLED=false`.



## Improved planning board

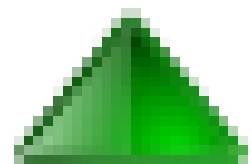
You can now quickly add or edit artifacts, know the card count and planning folder capacity of a selected planning folder and refresh individual swim lanes to see very recent artifact updates without leaving the planning board. In addition, when you expand an artifact, the parent and child artifacts are visually outlined by means of colored boxes. For more information, see:

- [Set up the planning board](#)
- [Work with artifact cards](#)



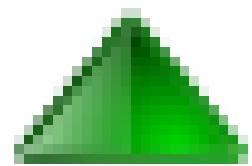
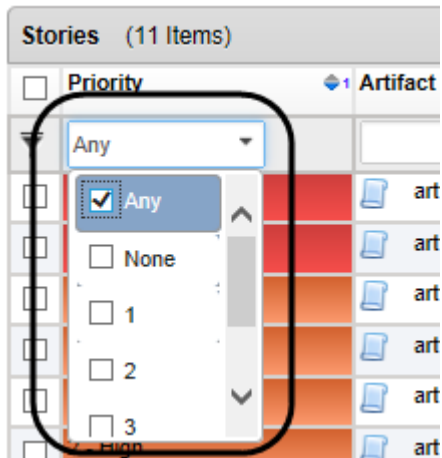
The screenshot shows a window titled 'PF2' with a toolbar containing a plus sign, a checkmark, and a refresh icon. The toolbar also displays '36 / 100 Pts' and '2 Cards'. The main content area lists four artifacts, each with a minus sign, a document icon, a title, a priority level in parentheses, an edit icon, points, a person icon, and an 'Open' button. The artifacts are: 'artf1110 (P2)' with 24 points, 'CTF 7.1 Refresh Swimlanes'; 'artf1111 (P2)' with 12 points, 'CTF 7.1 Table Filtering'; 'artf1112 (P4)' with 0 points, 'Multi-select drop-down lists'; and 'artf1113 (P4)' with 0 points, 'Lucene upgrade to v4.4'. The first three artifacts are grouped together in a blue-bordered container, while the fourth is in a separate white-bordered container.

Artifact ID	Priority	Points	Title	Assignee	Status
artf1110	P2	24	CTF 7.1 Refresh Swimlanes	None	Open
artf1111	P2	12	CTF 7.1 Table Filtering	None	Open
artf1112	P4	0	Multi-select drop-down lists	None	Open
artf1113	P4	0	Lucene upgrade to v4.4	None	Open



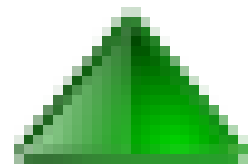
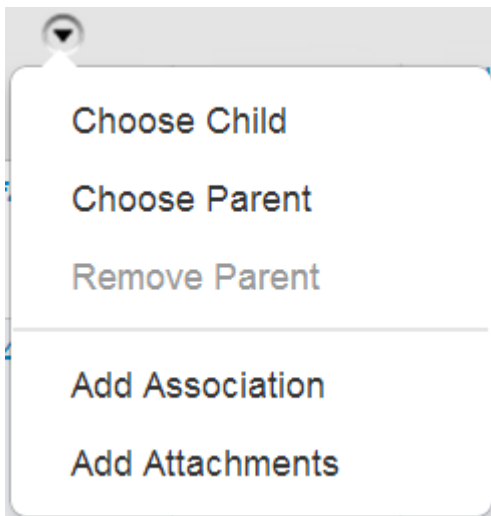
## Multi-select drop-down lists for filtering

You can now filter rows in less time and with fewer clicks by selecting more than one filter value in all table filter drop-down lists. For more information, see [Filtering tables](#).



### Context menu to quickly create dependencies, associations and add attachments

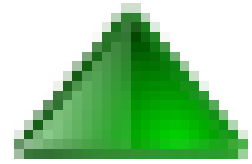
Choose a parent or child artifact, remove a parent, associate artifacts and add attachments to artifacts in less time and with fewer clicks. For more information, see [Quickly create dependencies, associations or add attachments](#).



### Extended path-based permissions with partial wildcard match

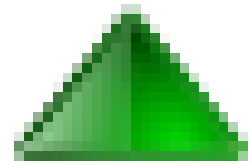
You can now write rules that partially uses wildcards in file or folder names such as: `/trunk/build/*.iso`. For more information, see:

- [Who can access source code?](#)
- [Wildcard-based access control and path-based permissions in TeamForge](#)



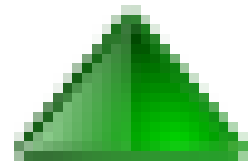
### System tool to change a site's log level

You can now use the **Configure Logging** page to change the application server (JBoss) log level without restarting the site. For more information, see [Configure your site's log level](#).



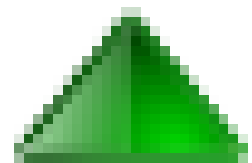
### Responsive drop-down and scrollable menus for mobile devices

TeamForge 7.1 has improved support for mobile devices.



### Site-wide role based permission for administrating Review Board

Site administrators can now grant a new permission, **Admin staff**, to users through a site-wide role. Users with both Admin staff and Submit/Edit/View permissions can access Review Board administration user interfaces. For more information, see [How do I control user access for Review Board?](#).

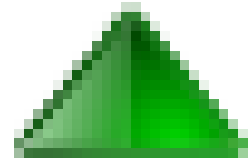


### Change tracker type during artifact edits

You can now change an artifacts tracker type on the **Edit Artifact** page itself. Note that if the tracker definitions differ, data could be lost after changing the tracker type. Depending on the nature of update, the artifact update

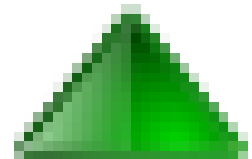
success message clearly conveys whether you have updated the title or description of an artifact or whether you have just changed an artifact's tracker type. For more information, see:

- [Edit a tracker artifact](#)
- [Is it possible to move an artifact from one tracker to another?](#)



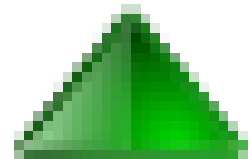
### Highcharts-based interactive charts

The Flash Player-based **Open by priority** and **Open Vs Closed** charts on the tracker and planning folder **List Artifacts** pages are replaced by Highcharts-based interactive charts.



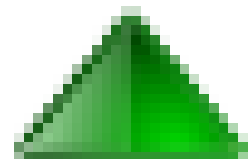
### Multi-threaded search indexing

TeamForge 7.1 uses Lucene 4.4, which has major improvements w.r.t indexing speed. To leverage Lucene's indexing speed improvements, TeamForge 7.1 is equipped with multi-threaded indexing process.



### New life cycle metric charts

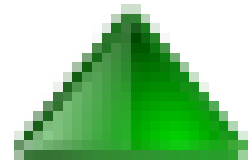
Two new life cycle metric charts are available: Release Burn Up Chart and Release Burn Down Chart. Check out the [Life Cycle Metric Charts](#) section for more information.



### Software version upgrades

See the [Software requirements for CollabNet TeamForge 7.1](#) on page 356 topic for a list of software tested for compatibility with TeamForge 7.1.

- **Lucene** upgraded to version 4.4: See [Upgrade TeamForge 7.1 search index to Lucene 4.x format](#) on page 318 for more information.
- **Subversion** upgraded to 1.8.3
- **JDK** upgraded to 1.7.0\_40
- **Review Board** upgraded to 1.7.17
- **PostgreSQL** upgraded to 9.2.4



## Issues resolved in CollabNet TeamForge 7.1

---

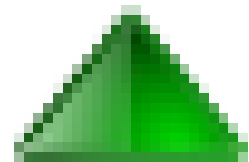
TeamForge 7.1 has a number of security vulnerabilities fixed, including a few around XSS.

Here's a list of few note-worthy issues fixed in TeamForge 7.1.

- *Security fixes.*
- *Inline-editing of the estimated, remaining and actual effort fields is no more allowed if the **Calculate Effort: Sum effort from children** check box is selected for the artifact.*
- *A few parsing issues with inline editing were fixed.*
- *Random page movements issue during inline editing has been fixed.*
- *A new configurable Subversion timeout token is added to configure timeout values for busy sites.*
- *TeamForge 7.1's indexing process has been fine-tuned to prevent out of memory errors (OOM).*
- *A new site options token, LISTEN\_BACKLOG, is added to prevent SYN flooding.*

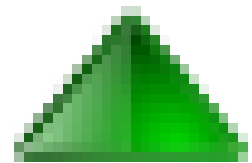
### Security fixes

Security vulnerabilities, especially around XSS, have been fixed across the application.



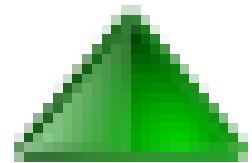
### Inline editing-related fixes

- In TeamForge 7.1, inline-editing of the estimated, remaining and actual effort fields is not allowed if the **Calculate Effort: Sum effort from children** check box is selected for the artifact.
- When you edit artifacts inline, you may have witnessed some parsing issues with fields such as "Fixed in Release" and "Reported in Release". This is fixed in TeamForge 7.1.
- During inline editing of artifacts, you may have witnessed 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. This is fixed in TeamForge 7.1.



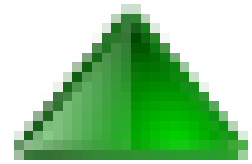
### Configurable Subversion timeout token

The "mod\_authnz\_ctf" module had a hard coded timeout of 30 seconds for soap calls, which resulted in Subversion operation failures in busy sites. This is fixed in TeamForge 7.1. A new configurable token, [SYN\\_AUTHNZ\\_TIMEOUT](#), with a default value of 60 seconds, is added. You may increase the timeout value for busy sites.



### Indexing-related fixes

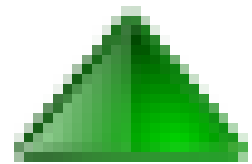
- When a new comment or an attachment (or both) is added to TeamForge work items (such as artifacts, wiki pages, tasks and so on) that already have a huge number of comments or attachments, an out of memory error (OOM) may occur while indexing the items. This issue has been fixed and the indexing speed has considerably improved.
- When multiple search operations are performed simultaneously on sites with huge volume of data, an out of memory error (OOM) occurred in the Phoenix server. This issue has been fixed. The search and indexing speed has considerably improved.



### A new token to prevent SYN flooding

It was found that the Apache server was responding slowly due to SYN flooding. To resolve this, you must add the ListenBackLog Apache setting to the value returned by "sysctl net.ipv4.tcp\_max\_syn\_backlog". For more information, see [this page](#).

You can control the value of this Apache setting (ListenBackLog) by setting the [LISTEN\\_BACKLOG](#) on page 401 `site-options.conf` token.





## Known issues in CollabNet TeamForge 7.1

---

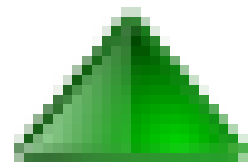
The following issues, including any workarounds we may have, are known to exist in the TeamForge 7.1 release.

List of known issues.

- *When a tracker field is disabled, it is no more visible in the UIs. However, its data is not deleted from the database (operational database and datamart).*
- *The **Open by priority** and **Open Vs Closed** chart components that are added in earlier versions of TeamForge as **free-form HTML components** will be blank post upgrade to TeamForge 7.1.*
- *For a given release, the Release Burn Down Chart does not exclude deleted sprint planning folders from the report.*
- *Run the TeamForge install command (recreate runtime) again if you encounter any JBoss exception during "recreate runtime".*
- *Rank mode may not display artifacts in the planning folder **List View**.*
- *Core Review Board bug prevents users from adding comment to files attached to review requests.*
- *Non-secured linked applications may not be accessible in SSL-enabled TeamForge.*

### Inconsistent data in datamart when tracker fields with data are disabled

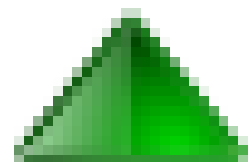
When you disable a tracker field, the field is removed from the user interface, but its data continues to live in the operational database and therefore in datamart database as well. This results in data inconsistency between what you may see in the user interface and what is actually available in the database.



### Flash-based Open by priority and Open Vs Closed charts

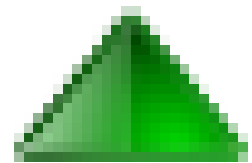
Flash Player-based charts are replaced by Highcharts-based interactive charts in TeamForge 7.1. As a result, after upgrading to TeamForge 7.1, the **Open by priority** and **Open Vs Closed** chart components that are added in earlier versions of TeamForge as **free-form HTML components** will be blank.

Post upgrade to TeamForge 7.1, instead of adding the charts again as free-form HTML components, add the required charts (**Open by priority** and **Open Vs Closed**) as tracker metric chart components. See [Create a project page component](#) to know how to add a **Tracker Metrics** chart component to your project home page.



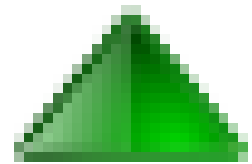
### Release Burn Down Chart

For a selected release planning folder, the Release Burn Down Chart shows data from all sprint planning folders, even if one or more sprint planning folders are deleted in due course.



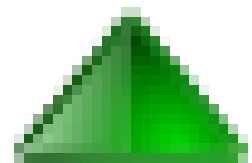
### **JBoss exception during "recreate runtime"**

A JBoss exception was encountered during TeamForge installation (during "recreate runtime"). This could be an issue with the JBoss vault library and is found to occur inconsistently. If you encounter this exception during installation, run the TeamForge install command (recreate runtime) again.



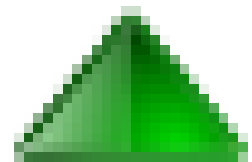
### **Rank mode may not display artifacts in the planning folder List View**

If you have any of the parent or child artifacts expanded in the planning folder sort mode and then click **Rank**, the "Rank" mode may not display artifacts in the planning folder **List View**. As a workaround, you may use the planning board for ranking artifacts.



### **Core Review Board bug prevents users from adding comments to files attached to review requests**

When you post a review request with file attachments, other users may not be able to add comments to the files attached to the review request. This is a core Review Board bug.



### **Non-secured linked applications may not be accessible in SSL-enabled TeamForge**

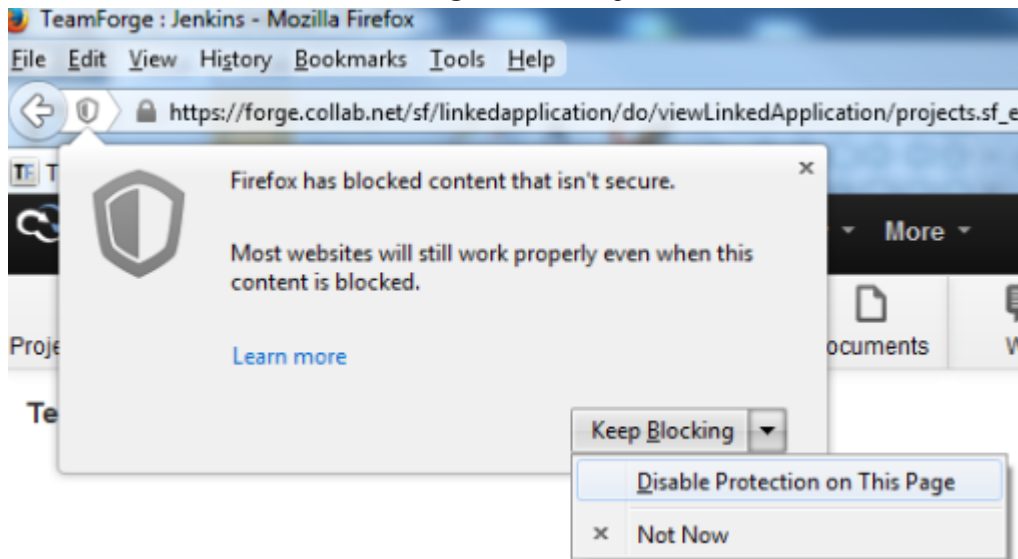
If you run TeamForge in SSL mode (https), you may not be able to access linked applications such as Jenkins that run in a non-SSL mode (http).

#### **Workarounds:**

- **Internet Explorer:** While trying to access the linked application, click **Show all content** when the browser warning message appears.

Only secure content is displayed. [What's the risk?](#)

- **Firefox:** While trying to access the linked application, click the security shield icon on the browser address bar and click **Disable Protection on This Page** from the drop-down list.



- **Chrome:** While trying to access the linked application, click the security shield icon on the browser address bar and click **Load unsafe script**.

