

CollabNet Subversion Edge User Guide

Contents

Welcome to CollabNet Subversion Edge.....	7
Set up Subversion Edge	8
Install Subversion Edge	8
Install Subversion Edge on Windows.....	8
Install Subversion Edge on Linux.....	9
Install Subversion Edge on Solaris.....	10
Configure the server.....	12
Configure general server settings.....	12
Configure advanced server settings.....	13
Configure authentication settings.....	13
Configure a proxy server.....	14
Configure the mail server.....	14
Configure log settings.....	15
Configure server monitoring.....	15
Upgrade Subversion Edge	16
Update Subversion Edge from the console.....	16
Upgrade Subversion Edge manually.....	16
Manage repositories.....	17
Create a repository.....	17
Connect an existing repository.....	17
Define repository access rules.....	18
Schedule repository backups.....	19
Dump a repository.....	20
Load a repository.....	22
Verify repository data.....	23
Create a repository template.....	23
Manage hook scripts.....	24
Monitor the disk space used by a repository.....	24
Delete a repository.....	24
Manage users	26
Create a Subversion Edge user account.....	26
Edit a user account.....	27
Edit a role.....	27
Set up Subversion Edge as a TeamForge server.....	28
Convert to a TeamForge SCM integration server.....	28
Get the TeamForge Integration API Key.....	30
Import TeamForge's self-signed SSL certificate.....	30
Convert to standalone mode.....	31
Replication.....	32
Overview.....	32

Convert to a replica server in TeamForge.....	33
Monitoring a replica server.....	39
Update TeamForge account for replica mode.....	40
Update replica retry settings.....	41
Convert to standalone mode.....	41
Status icons for replica jobs.....	42
Extend Subversion Edge with the CollabNet cloud.....	43
What is a cloud backup?.....	43
Schedule a cloud backup.....	43
Access a cloud backup.....	44
Restore a repository from a cloud backup.....	44
Maintain your Subversion Edge site.....	46
Monitor server health.....	46
Monitor server logs.....	46
Monitor jobs.....	47
Subversion Edge FAQ.....	48
Can I use the web console with my custom Apache configuration?	48
When I try to browse repositories I am getting an Internal Server Error. How do I fix this?	48
How do I migrate from my existing Subversion installation to Subversion Edge?.....	48
Using Subversion with TeamForge.....	48
Can I use a Subversion 1.7.x client with my TeamForge 6.1.1 server?	48
Can I upgrade the Subversion server installed by TeamForge to version 1.7.x?	48
Can I install Subversion 1.7.x on a Subversion Edge server set up as a TeamForge integration server?.49	49
Are there any other upgrades I need to consider when upgrading to Subversion 1.7.x?	49
.....	49
Can I install Subversion 1.7.x on a Subversion Edge server set up as a TeamForge replica server?.49	49
Subversion Edge REST API.....	50
General usage.....	50
Subversion Edge User API.....	52
Subversion Edge User Roles API.....	53
Subversion Edge Repository API.....	54
Subversion Edge Repository Template API.....	55
Subversion Edge Repository Hook Scripts API.....	56
Subversion Edge Logging API.....	58
Subversion Edge Secure Port API.....	59
Reference.....	61
Supported software.....	61
Subversion Edge components.....	62
Scripts.....	62
svn.....	62
svn-httpd.....	62
Subversion Edge Release Notes	63
CollabNet Subversion Edge 4.0.4.....	63
Features and enhancements.....	63

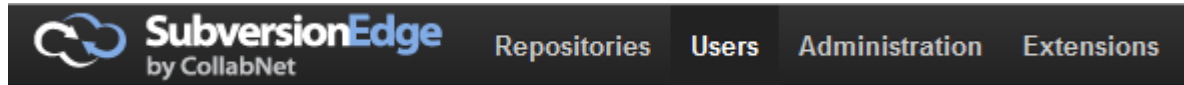
CollabNet Subversion Edge 4.0.3.....	63
Features and enhancements.....	63
Resolved issues.....	63
CollabNet Subversion Edge 4.0.2.....	64
Features and enhancements.....	64
Resolved issues.....	64
CollabNet Subversion Edge 4.0.1.....	64
Features and enhancements.....	64
Resolved issues.....	64
CollabNet Subversion Edge 4.0.0.....	65
Features and enhancements.....	65
Resolved issues.....	66
CollabNet Subversion Edge 3.3.2.....	66
Features and enhancements.....	66
Known Issues.....	66
CollabNet Subversion Edge 3.3.1.....	66
Resolved issues.....	66
Known Issues.....	67
CollabNet Subversion Edge 3.3.0.....	67
Features and enhancements.....	67
Resolved issues.....	67
CollabNet Subversion Edge 3.2.2.....	68
Features and enhancements.....	68
Resolved issues.....	68
CollabNet Subversion Edge 3.2.1.....	69
Resolved issues.....	69
CollabNet Subversion Edge 3.2.....	69
Features and enhancements.....	69
Resolved issues.....	70
CollabNet Subversion Edge 3.1.2.....	70
Resolved issues.....	70
CollabNet Subversion Edge 3.1.1.....	70
Added features.....	70
Resolved issues.....	70
CollabNet Subversion Edge 3.1.0.....	71
Added features.....	71
Resolved issues.....	71
CollabNet Subversion Edge 3.0.1.....	71
Added features.....	71
Resolved issues.....	72
CollabNet Subversion Edge 3.0.0.....	72
Added features.....	72
Resolved issues.....	73
CollabNet Subversion Edge 2.3.0	73
Added features.....	73

Resolved issues.....	74
CollabNet Subversion Edge 2.2.1	74
Resolved issues.....	74
CollabNet Subversion Edge 2.2.0	75
Added features.....	75
Resolved issues.....	75
CollabNet Subversion Edge 2.1.1	76
Added features.....	76
Resolved issues.....	76
CollabNet Subversion Edge 2.1.0	76
Added features.....	76
Resolved issues.....	77
CollabNet Subversion Edge 2.0.1	77
Added features.....	77
CollabNet Subversion Edge 2.0.0	78
Added features.....	78
Resolved issues.....	78
CollabNet Subversion Edge 1.3.3	79
Added features.....	79
Resolved issues.....	79
CollabNet Subversion Edge 1.3.2	80
Added features.....	80
Resolved issues.....	80
CollabNet Subversion Edge 1.3.1	80
Added features.....	80
Resolved issues.....	81
CollabNet Subversion Edge 1.3	81
Added features.....	81
Resolved issues.....	81
Known Issues.....	83
CollabNet Subversion Edge 1.2.2	83
Added features.....	83
Resolved issues.....	83
CollabNet Subversion Edge 1.2.1	83
Resolved issues.....	83
CollabNet Subversion Edge 1.2	84
Added features.....	84
Resolved issues.....	84
Known Issues.....	85
CollabNet Subversion Edge 1.1.0	85
Added features.....	85
Resolved issues.....	85
CollabNet Subversion Edge 1.0.1	86
Resolved issues.....	86
CollabNet Subversion Edge 1.0.0	86

Highlights of Subversion Edge 1.0.0	86
Known Issues.....	87

Welcome to CollabNet Subversion Edge

CollabNet Subversion Edge is a Subversion server distribution that includes a complete software stack of Apache, Subversion and ViewVC as well as a powerful web-based console for managing the server.



Project

The Subversion Edge project is located at <https://ctf.open.collab.net/sf/projects/svnedge/>.

What's new?

Subversion Edge 4.0.4 is the latest release. The release notes are available [here](#).

Related Links

[Introductory video on Subversion Edge](#)

Set up Subversion Edge

Setting up CollabNet Subversion Edge involves installing it and then configuring the Apache server using the web management console.



Note: Before you get Subversion Edge, keep in mind the following:

- Subversion Edge repositories are served via Apache httpd. There is no support for `svnserve` or SSH.
- Only the `fsfs` Subversion repository format is supported. There is no support for the BerkeleyDB (BDB) format.

Install Subversion Edge

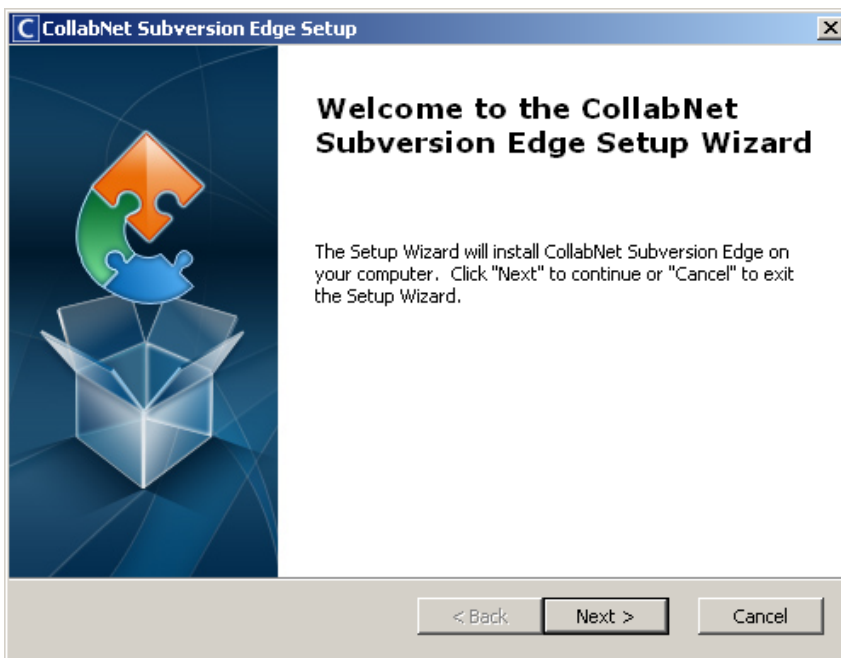
There is a standard MSI-based Windows Installer, and `tar.gz` archives for Linux and Solaris.

Install Subversion Edge on Windows

To install Subversion Edge, follow the steps in the installer and start the web console.

You must have Java 1.6 JRE/JDK installed. If the installer does not detect Java 1.6, it will install Java 6. It will create the `JAVA_HOME` environment variable if it does not already exist.

1. Get the installer from <http://www.collab.net/svnedge>.
2. Run `CollabNetSubversionEdge-x.x.x_setup.exe` and follow the steps in the installer.

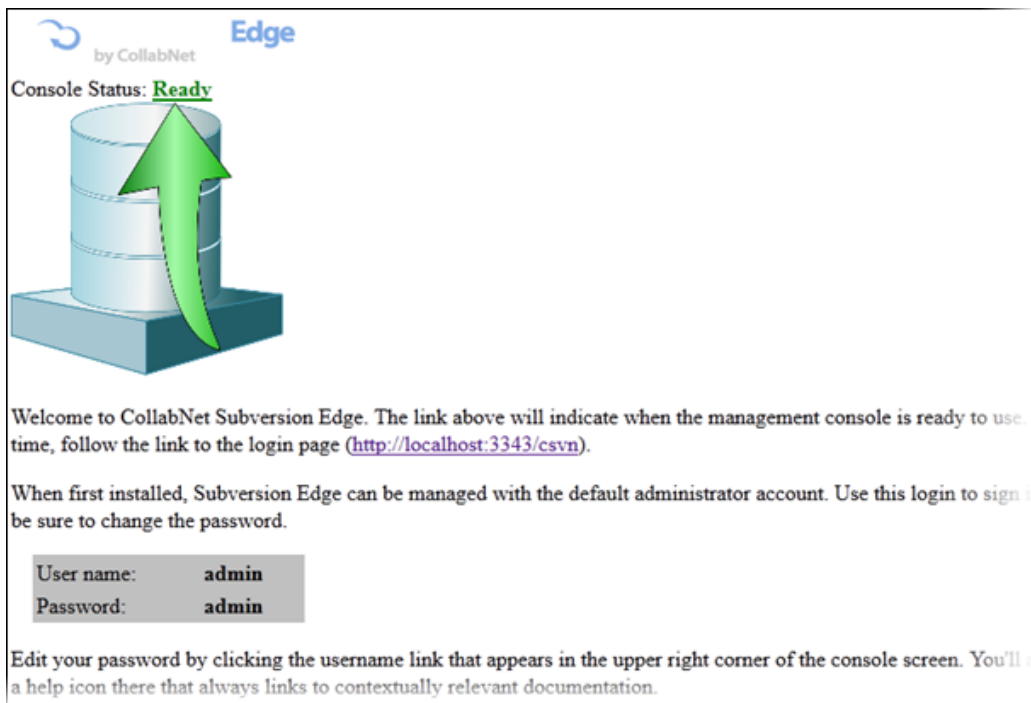


The installer updates the system `PATH` variable to include the path to the `bin` and `Python25` folders that it installs. It also creates or updates the `PYTHONPATH` environment variable.

The installer also adds rules to the Windows Firewall to allow access to the Apache binary and to open up ports 3343 and 4434.

The installer adds these two Windows services, and sets them to automatically start when the system starts.

- CollabNet Subversion Server
 - CollabNet Subversion Edge
3. You have the option to start the application at the end of the install. If you select it, your browser will be opened to a local page containing a link to the console when the server has started.



4. Log into the console (at <http://hostname:3343/csvn>) with these default administrator credentials.
- Username: admin
 - Password: admin
5. Once you've logged in, use the Getting Started wizard to do the following:
- a) Change the password for the admin account by clicking the user name in the upper right corner of the screen and updating the information in the **Edit User** page.
 - b) *Configure the server.*
 - c) *Connect to CollabNet CloudForge.*

Related Links

[Additional information on running Subversion Edge on Windows](#)

Install Subversion Edge on Linux

The Linux distribution is a tar.gz archive. Expand it to the location where you want to install.



Important: You must install the application as a non-root user. If you install as a root user, you will later need to change the ownership of all files to a non-root user, and run the application as that non-root user.

1. Prepare the server by installing the required software.
 - Install a Java 6 JRE. The Sun/Oracle JRE is recommended.
 - Set the JAVA_HOME environment variable, and point it to your Java 6 JRE home. For example:


```
export JAVA_HOME=/usr/lib/jvm/java-1.6.0-sun
```
 - In case you do not have Python already, install it.

2. Switch to the parent directory where you want to install Subversion Edge.

```
cd /u1
```

3. Get the installer from <http://www.collab.net/svnedge>.
4. Unpack the installer.



Note: Make sure you extract the package as a non-root user.

```
tar -zxf CollabNetSubversionEdge-1.1.0_linux-x86.tar.gz
```

The tar.gz file is unpacked to the `csvn` directory.



Tip: You can rename the folder to whatever you want using the "mv" command.

5. Run the application.



Note: You must run the application as a non-root user.

```
cd csvn
```

```
bin/csvn start
```

It'll take a minute for the application to start. You should be able to access it from a web browser at `http://hostname:3343/csvn`.

6. Log into management console with these default administrator credentials.
 - Username: admin
 - Password: admin
7. Once you've logged in, use the Getting Started wizard to do the following:
 - a) Change the password for the admin account by clicking the user name in the upper right corner of the screen and updating the information in the **Edit User** page.
 - b) *Configure the server*.
 - c) *Connect to CollabNet CloudForge*.

Related Links

[Additional information on running Subversion Edge on Linux](#)

Install Subversion Edge on Solaris

The Solaris distribution is a tar.gz archive. Unzip and expand it to the location where you want to install Subversion Edge.



Important: Do not extract the installer using root or sudo. This will cause the UID/GID on the files to carry the values from our build system instead of being reset.

1. Prepare the server by installing the required software.
 - Install a Java 6 JRE. The Sun/Oracle JRE is recommended.
 - Set the JAVA_HOME environment variable, and point it to your Java 6 JRE home. For example:

```
$ export JAVA_HOME=/usr/java
```

- Test the variable:

```
$ $JAVA_HOME/bin/java -version
```

```
java version "1.6.0_20" Java(TM) SE Runtime Environment
build 1.6.0_20-b02) Java HotSpot(TM) Client VM (build 16.3-b01,
mixed mode, sharing)
```

- You must have Python 2.4 to 2.6 installed. We tested with the 2.4 version provided with Solaris. If you use Python from another source, it must be compiled with shared libraries.

2. Switch to the folder where you want to install Subversion Edge.



Note: You must have write permissions to this folder.

```
$ cd /opt
```

3. Get the installer from <http://www.collab.net/svnedge>.

Separate downloads are available for SPARC and x86. Get the one appropriate for your Solaris architecture.

4. Unzip and expand the file.



Note: Do not untar CollabNet Subversion Edge using root or sudo.

```
$ gzip -d CollabNetSubversionEdge-x.y.z_solaris-x86.tar.gz
$ tar xf CollabNetSubversionEdge-x.y.z_solaris-x86.tar
```

This will create a folder named `csvn` in the current directory. You can rename the folder to whatever you want.

5. This step is optional. To install the application so that it will start automatically when the server restarts, enter these commands:

```
$ cd csvn
$ bin/csvn install
```



Note: Do not use sudo to run this command. Run it as the user you want to run the server. The command script will in turn use sudo to install the application, so you may be prompted for the password.

In addition to configuring your system so that the server is started with the system, the installer also writes the current `JAVA_HOME` and the current username into the file `data/conf/csvn.conf`. You can edit this file if needed as it controls the startup settings for the application. By setting the `JAVA_HOME` and `RUN_AS_USER` variables in this file, it ensures they are set correctly when the application is run.



Note: The install script will use sudo to create the necessary symlinks to autostart the application. You may be prompted by sudo for your password.

6. This step, for configuring a proxy server, is optional. Subversion Edge needs access to the internet to check for and install updates. If you need to go through a proxy to access the internet, you can configure the proxy by editing the `data/conf/csvn.conf` file which was created by the previous step -- uncomment and edit the `HTTP_PROXY` variable.

7. Start the server.



Note: Be sure that you are logged in with your own userid and not running as root.

```
$ cd csvn
$ bin/csvn start
```

This will take a few minutes and the script will loop until it sees that the server is running. If the server does not start, then try starting the server with this command: `$ bin/csvn console` This will start the server but output the initial startup messages to the console.

8. Log into the management console at `http://hostname:3343/csvn` with these default administrator credentials.
 - Username: admin
 - Password: admin
9. Once you've logged in, use the Getting Started wizard to do the following:
 - a) Change the password for the admin account by clicking the user name in the upper right corner of the screen and updating the information in the **Edit User** page.
 - b) [Configure the server](#).
 - c) [Connect to CollabNet CloudForge](#).

Related Links

[Additional information on running Subversion Edge on Solaris](#)

Configure the server

Once the installation is done, you need to set up the Apache server that hosts the Subversion repositories.

Administrators with the `ROLE_ADMIN` or `ROLE_ADMIN_SYSTEM` role can configure the server.

Configure general server settings

The Apache configuration files get written based on the information you provide in the web console. You don't need a direct login to the host server operating system.

1. Click **Server Settings** as the second step of the Getting Started wizard or from the left panel after selecting **Administration** in the main navigation bar.
2. The following fields are provided values. Make changes as appropriate.
 - **Hostname:** This is the fully qualified hostname including the domain of your Subversion Edge Apache server. For example, `cu001.cubit.abc.collab.net`.
 - **Port:** For Windows, the default is the standard port 80. On Linux, you need to perform some additional setup to use a standard port such as port 80, or port 443 if you want to turn on SSL and use HTTPS. When you select a reserved port, examples of what you need to do will be displayed. To run the server on any port less than 1024, you need read access. The system will check for this as well.
 - **Repository Directory:** This is the path to the parent directory for your Subversion Edge repositories. The default is `/data/repositories` in the location where you installed Subversion Edge. If you have existing repositories you want to connect to the web console, you could enter that location instead.
 - **Backup Directory:** Repository dump files are stored in this location under the name of the repository. The default is `/data/dumps` in the location where you installed Subversion Edge.
3. Specify whether you want encryption for the Subversion server.

If you enable this setting, you will be provided the **Advanced Configuration** link to set up custom SSL directives.
4. Provide the site administrator's name, email address, and alternative contact information such as a phone number or another email address.

Apache error messages print the administrator's name and email address.
5. Specify whether the Subversion Edge console should require HTTPS.

In release 1.2 and later, Subversion Edge is pre-configured to run SSL on port 4434 using a self-signed certificate, in addition to the existing HTTP on port 3343. If you set HTTPS as a requirement for the console, all requests on port 3343 will get redirected to port 4434. Users who'll be accessing Subversion Edge via SSL will need to accept the certificate warnings unless you purchase an official SSL certificate from a trusted Certificate Authority.


If you've configured Apache for SSL, it will be set to use the same self-signed certificate as the console.

6. Click **Save**.

You need to configure the [Authentication](#) settings next. Configuring [Advanced Settings](#) before that is optional.

Configure advanced server settings

Subversion Edge 4.0.0 provides a page to update advanced server settings that are defined in the configuration files. While most users would not need to touch these settings, the page provides an easier alternative to having to manually customize the configuration files.

1. Click **Server Settings** as the second step of the Getting Started wizard or from the left panel after selecting **Administration** in the main navigation bar.
2. In the **Server Settings** page, click the **Advanced Settings** link.
3.  **Note:** The fields on the **Advanced Server Settings** page have default values which are generally adequate for a Subversion server. For more information, see [this wiki page](#).

If your server requires different values, update those settings.

4. Click **Save**.

Configure authentication settings

Specify how you want to authenticate your Subversion Edge installation. You can authenticate against a local `htpasswd` file or against a corporate LDAP server.

1. Select **Administration** in the main navigation bar and click **Authentication**.
2. Select **Anonymous Access** to allow anonymous users to be able to check out repositories.
3. Specify whether you want usernames to be converted to lower case for repository access rules.
If you select **Convert Usernames to Lower Case**, you must enter all usernames in lower case while editing access rules in the **Repository Access Rules** page.
4. To authenticate against a local `htpasswd` file, keep the default setting for **Authentication Methods**.
5. Select LDAP authentication to authenticate against a corporate LDAP server. It is possible to select local as well as LDAP authentication. If both authentication methods are enabled, local authentication is tried first, and then LDAP authentication.

You should contact your LDAP administrator for the best ways to configure LDAP support. Here are some examples:

- **LDAP Security Level:** NONE establishes an unsecure connection on the default LDAP port. SSL establishes a secure connection on the default secure LDAP port. TLS/STARTTLS establishes an upgraded secure connection on the default LDAP port.
- **LDAP Server Host:** the IP address or hostname of your directory server. Example, `your.domain`
- **LDAP Server Port:** The default port 389 talks to the local directory server. To have Apache query the "Global Directory" on an Active Directory server, use port 3289.
- **LDAP Base DN:** the distinguished name to the path in the directory tree that you want to search for users. Example, `DC=your,DC=domain`
- **LDAP Bind DN and LDAP Bind Password:** the distinguished name and password of the user account that Apache uses to connect to the directory system to perform its user authentication. For example, `"CN=ldapuser,CN=Users,DC=your,DC=domain"` and `ldappassword`
- **LDAP Login Attribute:** the attribute used to identify the user's login name. Example, `sAMAccountName` attribute for Active Directory.
- **LDAP Search Scope:** tells the directory server how deep to query. Example, `sub`
- **LDAP Filter:** tells the directory server to filter the query for the specified objects. For example, `(objectClass=*)` searches for all objects; `(objectClass=user)` searches for objects of type "user".

6. Click **Save**.



Note: Since Subversion Edge 1.3, users can use their LDAP credentials to log into the Subversion server as well as the console. Upon first sign-in to the console, the LDAP user is created as a console user with `ROLE_USER` permissions. This user cannot be edited, but a Subversion Edge administrator can assign additional roles via the **Edit Role** screen to allow the user to administer the server using LDAP credentials.

Related Links

[Subversion with Apache and LDAP](#)

Configure a proxy server

If your Subversion Edge server requires a proxy for outbound connections, provide its configuration in the web console.

You would need to do this for certain functionality to work:

- To get automatic updates
- When Subversion Edge needs to go through a proxy to reach a TeamForge server, such as when acting as a Subversion replica

These settings do not affect inbound connections from web browsers or Subversion clients.



Note: For TeamForge modes:

- In TeamForge replica mode, we do not support a proxy that requires authentication. This is because the Apache server has to be configured to use the proxy and it does not support authentication.
- When used as a primary integration server in TeamForge, the Subversion Edge server can use a proxy to reach TeamForge, but must also be reachable from the TeamForge host.

1. Select **Administration** in the main navigation bar and click **Proxy Server**.
2. In the **Proxy Server Configuration** page, provide the host name or IP address of the proxy server and the port number.
3. Provide the username and password for the proxy server only if authorization is required.



Note: If the Subversion Edge server is used as a Subversion replica in TeamForge, proxy authorization is not supported.

4. Click **Save**.

Configure the mail server

To receive email notifications from the Subversion Edge server, set up the mail server with the details of your corporate email server. When a scheduled background job encounters a problem, an email with information from the error is sent to different accounts depending on the situation.

- Repository dump and load problems are sent to the email address of the user who submitted the job. The server administrator is copied on the email.



Note: The server administrator account is the email address specified in **Administrator Email** in the **Server Settings** page.

- Backup failures are sent to the email address of the server administrator.

When a dump or load job finishes, an email is sent to the user who started the operation. Success emails are not sent for backup dumps.

1. Select **Administration** in the main navigation bar and click **Mail Server**.
2. In the **Mail Server Configuration** page, select **Enable email notifications**.
 - a) Enter the hostname or address of the mail server. For example, `exchange.collab.net`.
 - b) Enter the port number of the mail server.

- c) If the mail server requires authentication, provide the username and password.
- d) Select the connection security protocol.
- e) Specify the From address to be used in emails.

If you don't provide a value, <username>@<mail.server.name> or SubversionEdge@<mail.server.name> is used.

3. Click **Test Settings** to verify your configuration.

The server will wait for up to 10 seconds for the mail server to send a test email to the address specified. If the test does not definitively succeed or fail within that time, you will see a response about the text taking longer to complete and have an option to cancel it. Eventually the page will be updated with the test result.

4. Click **Save.**

Configure log settings

Specify logging levels for the Subversion Edge console and the Apache Subversion server, how long you want to keep the logs, the maximum size for the logs you want to retain, and whether you want to turn off the Apache access or Subversion logs if you are not interested in their content and do not want them using disk space.

You need the `ROLE_ADMIN` or `ROLE_ADMIN_SYSTEM` role to be able to configure log settings.

- 1. Select **Administration**** in the main navigation bar and click **Logging** in the left panel.
- 2. Enter the number of days you want the logs retained.**
A value of "0" indicates that you don't want the logs deleted.
- 3. Set the log level for the Subversion Edge console application.**

This determines the verbosity of the messages recorded in the log files. In decreasing order of significance, the four available levels are ERROR, WARN, INFO and DEBUG. When you select a level, all messages for levels of higher significance are recorded as well.

- 4. Set the error log level for the Apache Subversion server.**
- 5. By default, Apache access logging is turned on. To turn it off, de-select **Access Log**.**
- 6. Select **Minimize Logging**** to reduce the logging noise that Subversion 1.8 clients can generate.
Subversion 1.8, which Subversion Edge 4.0.0 supports, uses Serf. Because of the number of HTTP GET requests that Serf clients issue, you can expect a growth in server logs. See <http://subversion.apache.org/docs/release-notes/1.8.html#neon-deleted> for more information.
- 7. By default, Subversion command logging is turned on. To turn it off, de-select **Subversion Log**.**
- 8. Specify the maximum size (in MB) for the logs you want to retain.**
A value of "0" indicates that there is no limit.
- 9. Click **Save**.**

Related Links

[Monitor server logs](#)

Configure server monitoring

Specify the settings for gathering network and file system statistics.

- Select **Administration** in the main navigation bar and click **Server Monitoring**.
 - a) Specify whether you want to gather network statistics.
 - b) Specify whether you want to gather repository disk usage statistics.
If yes, configure how often you want this done.



Note: Gathering disk space usage data can be resource intensive. You may want to run this just once a day or even turn it off.

Upgrade Subversion Edge

Updates give you the latest fixes and enhancements to Subversion Edge, as well as the underlying native components such as Apache, Subversion and ViewVC.



Tip: If you already have Subversion installed — CollabNet Subversion, open-source Subversion or some other Subversion server product — and want to upgrade to Subversion Edge, see the [wiki on migrating from an existing Subversion installation](#).

Update Subversion Edge from the console

We recommend you use the application's integrated mechanism to install updates from your browser.



Note:

- Do not download and run a new version of the application installer.
- If your setup uses a proxy server to connect to the Internet, [configure the proxy server](#) before you install updates.

When updates are available, you will see a notification.

1. To get new updates, so one of the following:

- Click the **download** link in the message about available updates.
- Select **Administration** in the main navigation bar and click **Software Updates**.

2. In the **Software Updates** page, click **Install Updates**.

You will see the progress of your upgrade. When the Subversion Edge packages are downloaded and the upgrade can be applied, you will see a **Restart Server** option.

3. Click **Restart Server** to complete the upgrade.

You should see a message that your updates were installed successfully.

Related Links

[Subversion Edge packages](#)

Upgrade Subversion Edge manually

There are scenarios where you might need to do a manual upgrade: for example, your Subversion Edge server might not be able to access the Internet to pull down the updates or maybe a critical installation file has become corrupted and you need to reinstall using the same version.

Follow [these instructions](#) to perform a manual upgrade or reinstall.

Manage repositories

CollabNet Subversion Edge provides a web interface to create and import repositories and manage them.

Create a repository

When you add a new repository, you can set it up using a template or a backup that you created earlier.



Note: To be able to create a repository, you must be a CollabNet Subversion Edge administrator.

1. Select **Repositories** in the main navigation bar.
2. In the **Repositories** page, click **Create**.
3. Provide a name for the repository.



Note:

- The name must be unique.
 - Alphanumerics (Latin and ASCII character sets), underscore, dash and period characters are allowed. Mixed case is allowed.
4. In the **Create Repository** page, select one of these options to initialize the repository.
 - **Template**
 - An empty repository whose layout you can configure any way you choose
 - The standard trunk/branches/tags tree structure for repository files and directories
 - A template created from a dump file or zip archive of a repository's contents
 - **Backup:** load the new repository from an existing backup.
 - **Cloud Repositories:** select a repository that you backed up to the *CollabNet Cloud*. It will be restored as a new local repository.
 5. Click **Create**.

The repository is created with the proper filesystem permissions on your Subversion Edge server.

Related Links

[Create a repository template](#)

[Schedule repository backups](#)

Connect an existing repository

To start managing your existing repositories with the web interface, you need to first make Subversion Edge aware of their location.

There are two ways to connect an existing repository:

- Provide the path to your repository while configuring CollabNet Subversion Edge.
 - Move your repository to the default parent directory location.
1. Depending on how you choose to connect your repositories, do one of the following:

- In the main navigation bar, select **Administration**. In the **Server Settings** page, provide the path to your repositories in the **Repository Directory** field.
- Manually move your repositories into the default parent repository directory - `/data/repositories` in the location where you installed CollabNet Subversion.

2. Select **Repositories** in the main navigation bar and click **Discover**.

Your existing repositories are listed. When you visit this page for the first time after a repository is discovered, you might see a message about fixing repository permissions on a Linux system. Subversion Edge requires that repository files and directories be writable, which may not be the case with imported repositories.

3. If filesystem permission changes are required, do the following:

- a) Select the repository and click **Info**.
You'll see an example of the command for the fix.
- b) Apply the fix and click **Validate Permissions**.

Define repository access rules

You can configure Subversion's native authorization rules to give users access to a whole repository, or to a specific path within a repository.

To edit repository access rules, you need the `ROLE_ADMIN` or `ROLE_ADMIN_REPO` role.



Note: Subversion 1.8 checks access rules before showing a list of repositories. If you do not have read access to a repository, you will no longer see it in the list. With Subversion Edge 4.0.0, the list of repositories displayed respects these rules as well.

1. Select **Repositories** in the main navigation bar.
2. Click **Access Rules**.
3. Click **Edit** and make your changes.



Note: The file is locked to other users until you click **Save** or **Cancel**.

- The format for an entry is `[repositoryname:/path/within/repos]`.

Example 1:

```
[dev:/branches/components/xxx]
user1 = rw
user2 = r
user3 =
```

This gives "user1" read and write access to the `/branches/components/xxx` directory in the "dev" repository and read-only access to "user2". User "user3" is denied access to this directory, since the username has neither "r" nor "w" assigned.

Example 2:

```
[ / ]
* = rw
```

This gives all users read and write access to all repositories.

- You can define groups in a section named `[groups]` and refer to groups within rules using `@groupname`.

Example:

```
[groups]
mgrs = user1, user2

[dev:/tags]
* = r
@mgrs = rw
```

This gives the users in the group "mgrs" read as well as write access to the "tags" directory in the "dev" repository. All other users have read access to this directory.

- Groups can contain other groups.

Example:

```
[groups]
mgrs = user1, user2
engrs = engr1, engr2
allgroups = @mgrs, @engrs
```

4. Click **Save**.

Related Links

[Path-based authorization in Subversion](#)

Schedule repository backups

As part of repository maintenance, you can schedule full dump or hotcopy backups and specify the number of backup files to retain. You can also back up your repositories to the CollabNet Cloud.

1. Select **Repositories** in the main navigation bar and click **Backup Schedule**.
2. In the **Backup Schedule** page, select the type of backup:
 - Cloud Services Backup -- this feature requires an account with CollabNet Cloud Service; when you register, you get 250 MB of free storage.
 - Full Dump Backup
 - Hotcopy Backup -- full backup including hook scripts and configuration files; faster than the dump options
3. In the **New Jobs** tab, select one or more repositories to back up.
4. Specify how often and at what times you want the repositories backed up.
5. Enter the number of backup files you want to retain.

You will see this option if you selected full dump or hotcopy backup. A value of zero will preserve all backup files.



Note: Dump files created on demand from the repository list are not affected by this setting. Only backups created by this schedule are cleaned up.

6. Select whether you want the **Use deltas** option.

This option is available for full dump or hotcopy backups. The advantage of selecting it is reduced file size. However, the dump files are more CPU-intensive to create and cannot be easily compressed or filtered.
7. Click **Create**.

In the **Administration > Jobs** page, you can see the active, scheduled, and completed backup jobs. Here's an example:

Finished Background Jobs					
#	Id	Description	Scheduled for	Started at	Finished at
1	RepoDump-repo1-fullDump-1332530240606	Backup or dump of repository 'repo1'. View result.	03/23/2012 12:17:21 PDT	03/23/2012 12:17:21 PDT	03/23/2012 12:17:21 PDT
2	RepoDump-repo2-fullDump-1332530777098	Backup or dump of repository 'repo2'. View result.	03/23/2012 12:26:18 PDT	03/23/2012 12:26:18 PDT	03/23/2012 12:26:18 PDT
3	RepoDump-repo1-hotcopy-1332544258603	Backup or dump of repository 'repo1'. View result.	03/23/2012 04:15:00 PDT	03/23/2012 04:15:00 PDT	03/23/2012 04:15:00 PDT
4	RepoDump-repo2-hotcopy-1332544258612	Backup or dump of repository 'repo2'. View result.	03/23/2012 04:15:00 PDT	03/23/2012 04:15:00 PDT	03/23/2012 04:15:00 PDT
5	RepoDump-repo1-fullDump-1332543646613	Backup or dump of repository 'repo1'. View result.	03/23/2012 04:20:00 PDT	03/23/2012 04:20:00 PDT	03/23/2012 04:20:00 PDT

Scheduled Background Jobs					
10 records per page		Filter: <input type="text"/>			
#	Id	Description	Scheduled for	Started at	Finished at
1	RepoDump-repo1-fullDump-	Backup or dump of repository 'repo1'.	03/23/2012 05:20:00 PDT	-	-

Backups created using the hotcopy or dump options are stored in the location specified in **Administration > Server Settings > Backup Directory**. The default location is `data/dumps` in the directory where you installed Subversion Edge.



Tip: With Subversion Edge 3.0.0, it is possible to schedule more than one backup job for a repository. For example, you can set up an hourly cloud backup, a daily hot copy backup, and a monthly repository dump.

You can also replace an existing job with the new schedule — on the **Existing Jobs** tab, select the job and click **Replace**.

Related Links

[Create a repository](#)

[Migrating Subversion repository data](#)

[svnadmin dump command](#)

[svnadmin hotcopy command](#)

Dump a repository

In addition to scheduled backups, Subversion Edge lets you perform an ad hoc dump of your repository. With this option, you can choose a range of revisions or use deltas for the dump, select a format for the dump file, and filter the dump results.



Note: To be able to dump a repository, you need the `ROLE_ADMIN` or `ROLE_ADMIN_REPO` role.

1. Select **Repositories** in the main navigation bar.
2. In the list of repositories, select the checkbox for the one you want to back up and click **Dump**. The **Create Repository Dump File** screen appears.
3. Select the options for the dump.

Revisions	You can <ul style="list-style-type: none"> • Specify a single revision • Enter a range of revisions, using a colon -- for example, LOWER:UPPER • Leave the field blank to include all revisions
Incremental	The first revision in the dump will contain only the files and directories modified in that revision. This is useful when you want to create a smaller dump file to load into another repository that already contains the data in the original repository.
Use deltas	The data is represented as a delta to previous revisions. While the advantage of this option is reduced file size, the dump files are more CPU-intensive to create and cannot be easily compressed or filtered.
Compress	By default, the dump result is compressed and stored as a .zip file.
Apply filter	This path-based filter limits the contents that are dumped. <ul style="list-style-type: none"> • Include: Paths matching the prefixes specified here are included in the dump file. • Exclude: Paths matching the prefixes specified here are not included in the dump file. • Drop empty revisions: If there are no changes to the repository as a result of filtering, those empty revisions are dropped from the dump file. • Renumber revisions: The revisions that remain after filtering are renumbered -- gaps which may result from using the Drop empty revisions option are eliminated. • Preserve revprops: Existing revision properties are preserved. If this option is not selected, an empty revision's properties will be removed, except for the date and log message. • Skip missing merge sources: Merge sources for paths removed by filtering are skipped.

4. Click **Dump**.

You will see a message that the dump process is running. When it completes, the dump file will be available in the location specified in **Administration > Server Settings > General > Repository Dump File Directory**. The default location is `data/dumps` in the directory where you installed Subversion Edge.

You'll see the dump files in the **Backup Files** tab. Here's an example:

Repository: repo1

Status:	OK	Filesystem Format:	FSFS version 4
Revisions:	1	Repository Format:	5
Size:	32.9 KB	Sharding:	Enabled (1,000 revisions)
Packed:	False	Representation Sharing:	True
Supports:	svndiff1, sharding, mergeinfo, rep-sharing, packed revs		
UUID:	9c854d39-e823-4a46-820d-11172adc4979		

[Hook Scripts](#)
[Backup Files](#)
[Scheduled Jobs](#)
[Size Report](#)

10 records per page Filter:

	File Name	Date	Size
<input type="checkbox"/>	repo1-r0_1-filtered-20120323121721.dump.zip	2012-03-23 12:17:21 PDT	478 bytes

Showing 1 to 1 of 1 entries

Related Links

[Load a repository](#)
[Create a repository template](#)
[svnadmin dump command](#)
[svndumpfilter](#)

Load a repository

Let's say you're moving your Subversion repository to a different system. You can create a new empty repository and load it with a dump file via the Subversion Edge console.

- On the server where you want to move your repository, create a new repository without the standard trunk/branches/tags structure.
- On your Subversion Edge **Repositories** page, select the checkbox for the repository you want to load, and click **Load**.
- Select the dump file to load into the repository.
You can choose a file with the standard dump format or a zip archive.
- Select whether you want to retain the current repository UUID, ignoring the value from the dump file.
In this example, since the repository is being moved from another system, leave the **Ignore UUID** option unchecked.
By default, when loading data into an empty repository, the repository's UUID is set to the UUID from the dump file. Selecting the **Ignore UUID** option will cause the UUID from the dump file to be ignored.
- Click **Schedule Load**.

The dump file gets scheduled to be loaded. You can monitor this job in the **Administration > Jobs** page page.

Related Links

[Dump a repository](#)

[svnadmin load command](#)
[Migrating Subversion repository data](#)
[More about the UUID option](#)

Verify repository data

To check the integrity of your repositories, you can verify them ad hoc or schedule a job that does this periodically.

This feature is available since Subversion Edge 3.0.0.

1. Select **Repositories** in the main navigation bar.
2. To verify a repository ad hoc, select it and click **Verify**.
3. To schedule a job to do this, select the repository and click **OK**.
 - a) In the **Scheduled Jobs** page, select the Verify Repository job.
 - b) Specify how often and at what times you want the repository verified.
 - c) Click **Create**.

You can check the status of the job in **Administration > Jobs**.

The Verify job runs the `svnadmin verify` command. The ad hoc option generates an email whether the verify succeeds or fails. The scheduled job generates an email only if the verify fails and there is corruption found.

The user who submitted the job is sent emails on success (for an ad hoc verify) as well as failure. In addition, the **Administrator Email** account specified in **Administration > Server Settings** is also sent emails on failure.



Note: To receive emails, you must select **Enable email notifications** and configure the settings in **Administration > Mail Server**.

Related Links

[svnadmin verify command](#)

Create a repository template

As an administrator, you can create templates from repository dump files or zip archives and use them to initialize new repositories.

These formats are supported for templates:

- A repository dump file
- Zipped archive of a repository dump file
- Zipped archive of the contents of a repository: this should contain the entire repository (including the `db` and `hooks` folders), not just a checkout of some portion of the repository

1. Select **Repositories** in the main navigation bar.
2. Select **Manage Templates** and click **Create** in the **Repository Template List** page.
3. On the **Create New Repository Template** page, provide a name for the template.
 Template names can contain up to 120 characters and include alphanumerics and spaces.
4. Browse to the zip or dump file you want to use for your template.
5. Click **Create**.

When you add a new repository, this template will be available for you to initialize the repository.

Related Links

[Create a repository](#)

[Dump a repository](#)

Manage hook scripts

Subversion Edge gives you several options to manage repository hook scripts.

You need the `ROLE_ADMIN_HOOKS` role to access this feature.

1. Select **Repositories** in the main navigation bar.
2. Select the checkbox for the repository you're interested in, and click **Info**.
The **Hook Scripts** tab displays a list of files in the repository's `hooks` folder. Files within sub-folders, if any, are not displayed.
3. You can manage your hook scripts using these options.
 - **Create**: upload a new hook script.
 - **Rename**: give a hook script a new name.
 - **Copy**: duplicate a hook script with a new name.
 - **Delete**: remove an individual hook script.
 - **Edit**: update the file in the editor. While you are making changes, the file is locked to other users until you click **Save** or **Cancel**.
 - **Download**: click the filename to download the hook script.

When you upload, rename or copy a hook script, the file is executable.

Related Links

[Hook scripts](#)

Monitor the disk space used by a repository

As the administrator for your Subversion Edge server, you may need to know the amount of disk space used by an individual repository and how the usage is growing over time.

By being able to see your largest repositories, you can also figure out if there is a specific repository you should move to another server, or possibly archive if it is dormant.

- To see disk space usage for an individual Subversion repository, click **Repositories** in the main navigation bar, select the checkbox for the repository and click **Info**.

To see the disk space used by all repositories, see the **Disk space chart** in the **Administration > Statistics** page.

Related Links


[Monitor server health](#)

Delete a repository

Let's say you have a project that was ended before it got off the ground. You might want to save disk space by deleting its source code repository.


To be able to delete a repository, you need the `ROLE_ADMIN` or `ROLE_ADMIN_REPO` role.

1. Select **Repositories** in the main navigation bar.

2. In the list of repositories, select the checkbox for the one you want to remove and click **Delete**. A confirmation dialog appears.
3.  **Note:** The delete operation cannot be undone. All data will be removed permanently.

To confirm, enter "I_AM_SURE" and click **OK**.

The repository is removed from the disk. All backup jobs scheduled for the repository are deleted.

-  **Note:** Any backups or dumps of the repository are retained, though no longer findable in the web console. To gain back disk space, you need to manually remove those -- the default location for dumps and backups is `data/dumps` in the directory where you installed Subversion Edge.

Related Links

[Dump a repository](#)

Manage users

Managing users in CollabNet Subversion Edge involves creating and managing user accounts and specifying what users can do using the web management console.

Create a Subversion Edge user account

To participate in a Subversion Edge site, team members must have a user account in the site. Root administrators and user administrators on the site can create these user accounts.

1. Select **Users** in the main navigation bar.
2. In the **Users** page, click **Create**.
3. On the **Create User** page, provide the user's full name, description, login name, password and email address.



Note:

- You must enter a unique **Login Name**. You can include alphanumeric and special characters other than `()\|'":\`&\"$,<>`
 - **Full Name** and **Password** are required as well.
 - Passwords are stored using the Apache `htpasswd` tool. Once added, users can login to the server to change their own password and information.
4. Enter the email address and provide a description.
 5. To delegate responsibilities to the team member, assign an appropriate role.

Here's a list of roles and actions permitted by each role.

Role	Permissions
ROLE_ADMIN	A root administrator has all permissions for the site. When you install CollabNet Subversion, the site has only this user.
ROLE_ADMIN_REPO	Repository administrators have permissions to <ul style="list-style-type: none"> • Create a repository • Connect an existing repository • Edit the Subversion authorization rules to give users access to a whole repository or any path within the repository
ROLE_ADMIN_SYSTEM	System administrators have permissions to manage the Subversion Apache server from the console. They can <ul style="list-style-type: none"> • Update server configuration options • Start and stop the server • Monitor logs
ROLE_ADMIN_USER	A user administrator, for example a team lead, has permissions to <ul style="list-style-type: none"> • Add (and remove) users • Activate user accounts • Update user profiles and passwords for all user accounts

Role	Permissions
ROLE_ADMIN_HOOKS	Hook script administrators can create, edit, rename, copy, download or delete repository hook script files.
ROLE_USER	This is the basic role required for an active user in the CollabNet Subversion site. However, even without this role, users can perform repository checkout and commit operations. Users with this role can edit their own password and profile information.



Note: You cannot grant another user permissions you don't have yourself.

6. Click **Create**.

The user is enrolled with the account and role you configured.

Edit a user account

As an administrator, you may need to reset a user's password or change the account status when the user has trouble accessing the site. You may also need to update the role when the user's responsibilities change.

Administrators with the `ROLE_ADMIN` or `ROLE_ADMIN_USER` role can edit all user accounts. Other users can only change their own password or profile information.

1. Select **Users** in the main navigation bar.
2. Click the username for the account you want to edit.
3. On the **Edit User** page, make your changes.
 - To remove the user account, click **Delete**.
 - If you are updating the user's role, you cannot grant permissions that you don't have yourself.
4. Click **Update**.

Edit a role

When you need to assign a role to multiple users, do that by editing the role.

Administrators with the `ROLE_ADMIN` or `ROLE_ADMIN_USER` role can edit roles.

1. Select **Users** in the main navigation bar and click **Role List**.
2. In the **Roles** page, click the role you want to edit.
3. Click **Edit** and make your changes. You can update the description and add or remove users who are assigned the role.



Note: You cannot add or remove your own account.

4. Click **Update**.

Set up Subversion Edge as a TeamForge server

To bring TeamForge features to Subversion Edge, convert your Subversion Edge server to a master integration server or a replica server in TeamForge.

You can convert Subversion Edge to function in TeamForge in either of these ways:

- As a master (core) software configuration management (SCM) integration server
- As a replica server that mirrors an existing integration server

The replication feature is available since version 2.0.0 to Subversion Edge administrators and requires TeamForge 6.1. For more information, see the [help on replication](#).



Note: The conversion to a master integration server in TeamForge requires a TeamForge site administrator account with permissions to create an SCM integration server. For creating a replica, you do not need to have a TeamForge site administrator account, but you do need full access to all the repositories involved.

Convert to a TeamForge SCM integration server

To convert to a TeamForge server, you'll need to provide the TeamForge site URL, administrator credentials and a project name. Existing Subversion Edge repositories and users are imported into this project.



Note: In Subversion Edge 2.0.0, port 4434 is configured to run SSL in addition to the existing HTTP on port 3343. When you convert your Subversion Edge server to TeamForge mode, the integrated server is configured for the port you're using when you do the conversion.

- If your Subversion Edge installation uses a self-signed SSL certificate, you must import that certificate into the Java keystore for TeamForge and restart the TeamForge server before the conversion can happen. See [these instructions](#) to import `svnedge.crt` into TeamForge's java keystore using the `keytool` utility.
- If the TeamForge server uses https and a self-signed SSL certificate, you need to import the server certificate into the Java keystore for the Subversion Edge console before you begin the conversion. See [this topic](#) for details.

1. Select **Administration** in the main navigation bar and click **TeamForge** under **Extensions**.
2. On the **Extend your Edge with TeamForge** page, click **Connect** for the **Federate your Subversion servers** option.
3. In the **TeamForge Credentials** page, enter these details:

- TeamForge server URL, for example, `http://teamforge.collab.net`
- Administrator credentials for the site



Note: This account must have permissions to create an SCM integration server on the TeamForge site.

- The value of the [Integration API Key](#) configured on the TeamForge server

TeamForge Server Credentials

1. TeamForge Credentials 2. TeamForge Project 3. TeamForge Users 4. Convert to TeamForge mode

Credentials for a CollabNet TeamForge Site Administrator are needed in order to add a new SCM integration server.

Server URL:
TeamForge server base URL including protocol and hostname.

Administrator Username:
A TeamForge administrator account which must have permission to add new scm integration servers.

Password:

Integration API Key:
If the CollabNet TeamForge server is available on a public network, then it should be configured to require a key for communication with the Subversion Edge server. Entering a value in this field is not setting the shared key, it must already be configured on the CollabNet TeamForge server.

- If using a hosted TeamForge installation, contact support to receive a key.
- Otherwise, consult the TeamForge configuration files under the property name: `sfmain.integration.security.shared_secret` to find a suitable key.

[Continue](#)

Click **Continue**.

4. On the **TeamForge Project** page, enter the name of the TeamForge project where you want the repositories on your Subversion Edge server added.

This can be a new or existing project. If the project doesn't exist, it will be added during the conversion process. Click **Continue**.

5. On the **TeamForge Users** page:

- Select **Import users** option to migrate the Subversion Edge users to TeamForge.
- To add the imported users as members of the Subversion Edge project you specified earlier, select **Assign membership**.

Click **Continue**.

6. On the **Convert to TeamForge mode** page, click **Convert**.

You will see the Subversion Edge server in the TeamForge site's **Admin > Integrations > SCM Integrations** page. Example:

The TeamForge project's **Source Code** page displays the imported repositories. Example:

Repositories and users are now managed by TeamForge.



Note: Users imported into TeamForge need to use the **Forgot your password** link to receive a ticket to set their new password.

TeamForge usernames and passwords are required to log in to the Subversion Edge console. TeamForge site administrators can see repository information, schedule backups and manage hook scripts. TeamForge users with site-wide SCM Edit permissions have access to **Administration** functionality such as installing updates and starting or stopping the server; however, they do not have access to repositories.

Related Links

[Import TeamForge's self-signed SSL certificate](#)

[Get the TeamForge Integration API Key](#)

[Convert to standalone mode](#)

[Get started with projects in CollabNet TeamForge](#)

[Source control in TeamForge](#)

[Manage users in TeamForge](#)

[Import a self-signed SSL certificate into the Java keystore for TeamForge](#)

Get the TeamForge Integration API Key

CollabNet TeamForge APIs use a shared secret key which the Subversion Edge server needs to communicate with the TeamForge site.

If your TeamForge installation is hosted, contact CollabNet Support to get this key. Otherwise, follow these steps:

1. Open the `sourceforge.properties` file located in `/runtime/conf` in your TeamForge site directory. For example, `/opt/collabnet/teamforge/runtime/conf/sourceforge.properties`.
2. Note the value of the property `sfmain.integration.security.shared_secret`.

Related Links

[Convert to a TeamForge SCM integration server](#)

[Convert to a replica server in TeamForge](#)

[Import TeamForge's self-signed SSL certificate](#)

Import TeamForge's self-signed SSL certificate

If your TeamForge server uses https and a self-signed SSL certificate, you may get a Java error while converting your Subversion Edge server to TeamForge. To resolve this, you need to import the server certificate into the Java keystore for Subversion Edge.

1. Get a copy of the TeamForge server certificate from your web browser.
 - a) Access your site using a web browser and double-click on the padlock icon (available in most browsers) to examine the server certificate.
 - b) Export the certificate to a file, and save it as `<server>.cer`.

There is usually an option somewhere in the resulting dialogs that lets you do this.

2. Locate the Java keystore.

This is at `jre/lib/security/cacerts` in the location where you installed Java.

3. Locate the Java keytool utility.

This is at `/bin/keytool` in the location where you installed Java.

4. Import the server certificate into the keystore.

On Linux:

- ```
$ export PATH=$JAVA_HOME/bin:$PATH
```
- ```
$ sudo keytool -import -trustcacerts -alias <server> -file <server>.cer -keystore $JAVA_HOME/jre/lib/security/cacerts
```

On Windows:

- `keytool -import -trustcacerts -alias <server> -file <server>.cer -keystore %JAVA_HOME%/jre/lib/security/cacerts`

For server, enter the hostname, not URL.

5. At the password prompt, enter `changeit`.

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

6. Restart the Subversion Edge console.

You should be able to connect to the TeamForge site to complete the conversion.



Note: If your Subversion Edge server uses a self-signed SSL certificate, you would need to import the certificate into the Java keystore for TeamForge and restart TeamForge services.

Related Links

[Convert to a TeamForge SCM integration server](#)

[Convert to a replica server in TeamForge](#)

[Get the TeamForge Integration API Key](#)

Convert to standalone mode

If you've converted Subversion Edge to a primary integration server or replica server in TeamForge, it is possible to revert to the standalone management console mode.

1. Select **Administration** in the main navigation bar and click **TeamForge Mode**.



Note: You'll see the **TeamForge Mode** link after a successful conversion to TeamForge.

2. On the **Convert to Standalone mode** page, supply your TeamForge administrator credentials.

This account must have permissions to remove integration servers in TeamForge.

3. Click **Convert**.

After the conversion, local authentication is used and console functionality is fully restored.

The console is put back as it was before you converted to TeamForge mode. Users and server settings that you had before the conversion still exist. Repository hook scripts are restored.

When you revert from a core TeamForge SCM integration server to standalone mode, what you see with repositories is based on what happened while you used TeamForge. If you created new repositories or deleted repositories, that's what you have after you convert to standalone mode. Repository content itself is not modified -- it is not rolled back to what it looked like before you converted to TeamForge mode.

When you revert from replica to standalone mode, you will once again be able to access the repositories you had before the conversion.

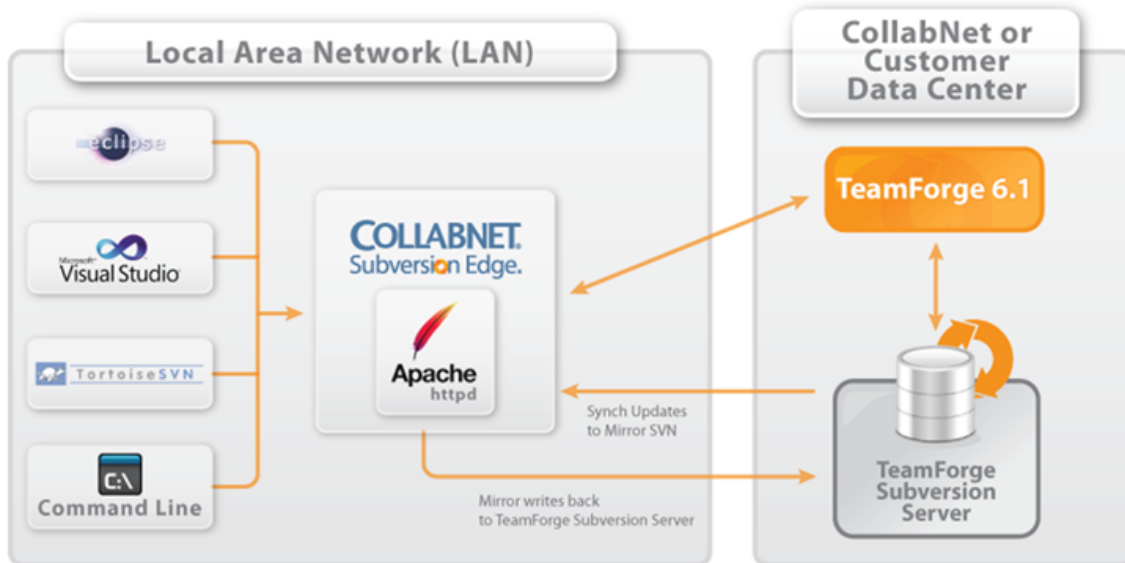
Related Links

[Convert to a replica server in TeamForge](#)

[Convert to a TeamForge SCM integration server](#)

Replication

TeamForge 6.1 and Subversion Edge 2.0.0 deliver improved performance to globally distributed users by enabling replication of Subversion repositories to local sites.



Replication is a wizard-driven process initiated in Subversion Edge. At the end of the process, the replica server is placed in a queue and needs approval by a TeamForge site administrator. When it is approved, you can make it available to TeamForge projects.

For any TeamForge project repository, there is a list of available replica servers. You can replicate the repository on a selected server.



Note: The TeamForge account used while setting up the replica determines what's available to be checked out from a replica repository.

When a replica server is deleted in TeamForge, it is reverted to a standalone Subversion Edge server.

Overview

Here's how Subversion Edge replication works, why replication could be useful in your setup, and what to consider when you plan to set up a replica.

Typically, you would deploy a replica for one of these reasons — to improve performance in remote project locations, and to reduce the load on the master server.

Projects in remote locations with lower bandwidth or higher latency want the performance of a "local" server.

Your company has a number of developers clustered together at a remote location. When you install a replica inside the LAN of these developers, they can greatly improve their Subversion performance and keep a lot of their traffic off the WAN. In this scenario, you would probably want a replica in each such location. Keep in mind that a replica can only be a proxy for one master — so if your company has more than one Subversion master server, you may need more than one replica at each location.

You want to reduce the load on the master server.

For example, continuous integration tools can place a lot of load on the server and moving that load to a separate server can increase the response time for other users. In this scenario you probably only need to add one replica; you'd add it as close to the master as possible so that synchronization is quick. Of course the previous point can be a factor here. If the continuous integration server is at remote location, then you would want to put a replica near the continuous integration server.

Rules for using a replica

When you *convert a Subversion Edge server to replica mode*, you provide the TeamForge username and password to use for the replica. The replica uses these credentials when it replicates Subversion content. This user must be added to the TeamForge project(s) and given permissions to the repositories being replicated. Those permissions also control what parts of the repository will be replicated. So if you have folders that should not ever live on remote servers, you can set up path-based permissions and that content will never be replicated to the server. If you forget to set up permissions, the replication will fail. However, there's no real harm done, and once you fix the permissions, you can do it again.

The replica user can be a normal user account — it does not have to be an Admin account. If the replica is set up and maintained by an Operations team, they might want to just use an Admin account so that project teams do not have to worry about adding the user to the project or setting up permissions.

Permissions for end users accessing those repositories will follow the normal TeamForge rules.

Architecture

All communication originates from the replica. TeamForge never contacts or pushes to the replica — the replica initiates this. When TeamForge wants the replica to do something such as synch a new repository or synch a new revision for an existing replicated repository, it queues an event for that replica. When a new commit comes in to TeamForge and a new commit object is created, the TeamForge application server queues up an event for each replica that has this repository. If the commit is for a repository with no replicas, then nothing happens.

The replica polls the TeamForge application server for its events. TeamForge site administrators can configure the polling interval for each replica individually. Typically the interval would be around 60 seconds. The longer the polling interval, the longer it will take for a commit to eventually reach a replica. At the same time, the longer the interval, the lesser the load placed on the application server by the polling mechanism.

The replica receives all of the queued events since the last time it polled, and splits these events into two groups:

- New repository initializations: Since these can take a long time, they are handled separately so as to not block all replication activity.
- All other transactions: This would mainly be synching revisions for existing replicas, but it could also be removing a repository or synching a revision property change.

TeamForge site administrators can control how many simultaneous jobs of each type run concurrently. Typically, the number for the first group would be low, such as 1 or 2. For the second group, it would be larger. Obviously the maximum simultaneous `svnsyncs` will have an impact on the Subversion master and the replica itself and must be considered.

For more information on replica settings and how to configure them, see *Edit replica settings*.

Related Links

[Convert to a replica server in TeamForge](#)

[Manage replicas in TeamForge](#)

Convert to a replica server in TeamForge

To convert your Subversion Edge server to a replica server in TeamForge, you'll need to provide the TeamForge site URL, a core SCM integration server to replicate, and administrator credentials.




Note: We recommend that you use a fresh install of Subversion Edge for conversion to a replica. If you have repositories in your Subversion Edge install, they will get archived in the `deleted-repos` directory after conversion -- you will not be able to access them in TeamForge. You will be able to keep your repositories only when you convert to a core SCM integration server in TeamForge.

1. Select **Administration** in the main navigation bar and click **TeamForge** under **Extensions**.
2. On the **Extend your Edge with TeamForge** page, click **Connect** for the **Optionally, replicate Subversion servers** option.
3. On the **TeamForge Credentials** page, provide the following information for your TeamForge site.
 - a) Enter the TeamForge server URL, for example, `http://teamforge.collab.net`.
 - b) Provide your credentials for the site.

**Note:**

- You need an account that's been used at least once in TeamForge.
- This account determines what's available to be checked out from a replicated repository in a TeamForge project. It can be set up in TeamForge to have access to the entire master repository, or with restricted access using path-based permissions.

- c) Enter the value of the *Integration API Key* configured on the TeamForge server.
- d)  **Note:** You will see the **Replica Sync Notification Address** field if you provided a value for **Administrator Email** in the **Server Settings** page and enabled email notifications.

Enter the email address to which notifications will be sent when replica synchronizations fail.

Log information will be attached to the email.

If you leave the field blank, emails will not be generated, but errors will still get logged.

- e) Click **Continue**.
4. On the **Replica Setup** page, do the following:
 - a) Select the TeamForge integration server you want to mirror.
 - b) Enter a description for the replica.
You might want to include the geographic location so that users can choose a nearby replica. For example, Brisbane, CA replica.
 - c) Enter a comment for the TeamForge Administrator.
For example, if you are in a large organization, you might want to identify yourself and why you need a replica.
 - d) Click **Continue**.
 5. On the **Convert to Replica Mode** page, make sure that all the information you've provided so far is accurate. If not, you can still make changes at this point. When you're done, click **Convert**.

TeamForge Replica Setup

1. TeamForge Credentials
2. Replica Setup
3. Convert to Replica Mode

You are ready! Verify the details of your replica. If changes are needed, revisit the previous steps. Click the 'Convert' button to switch to Replica Mode.

TeamForge Server:	https://cu345.cloud.sp.collab.net/
TeamForge Username:	admin
Master SVN Integration Server:	Subversion: <i>Internal Subversion Integration Server</i>
Description:	test server: <i>test server</i>
Message to TeamForge Administrator:	

Convert

The replica server in TeamForge will not be available for use immediately. It gets placed in a queue and needs approval by a TeamForge site administrator.



Note: If you supplied TeamForge site administrator credentials during the conversion, the replica server is automatically approved.

When approved, you will see the replica server in the TeamForge site's **Admin > Integrations > SCM Integrations** page.

Here's an example of a master Subversion server and its Subversion Edge replicas:

SCM Integrations				Pending SCM Integrations	Pending SCM Replicas	SCM Access Requests	Site-wide Linked Applications
Status	Name	Description		Category			
<input type="checkbox"/>	Subversion	Internal Subversion server		Subversion			
	London	Replica server in London office		Subversion Edge Replica			
	Sydney	Replica server in Sydney		Subversion Edge Replica			
	Beijing	Replica in Beijing office		Subversion Edge Replica			

Once a replica has been successfully registered with a TeamForge SCM server, it is available to project administrators in projects using that SCM server to house repositories. Project repositories are replicated when they are added to the replica server. See [these instructions](#) for details.

Related Links

[Import TeamForge's self-signed SSL certificate](#)
[Get the TeamForge Integration API Key](#)
[Update TeamForge account for replica mode](#)
[Overview](#)
[Convert to standalone mode](#)
[Get started with projects in CollabNet TeamForge](#)
[Source control in TeamForge](#)
[Approve a replica server in TeamForge](#)
[Replicate a repository in TeamForge](#)

Manage replicas in TeamForge

A replica server in TeamForge is a Subversion Edge server that replicates the content of an existing core SCM integration server. As an administrator, you can approve replica requests and configure replica settings.

Approve a replica server request

When there is a request for a replica of a core SCM integration server, a TeamForge administrator must approve the request before the replica is created.

Replica requests from a TeamForge admin user or site administrator are automatically approved. Replicas requested by other users need approval by a TeamForge administrator.

1. Click **Admin** in the CollabNet TeamForge navigation bar.
2. On the site administration navigation bar, click **Integrations**.
3. On the **SCM Integrations** page, click the **Pending SCM Replicas** tab.
4. From the list of pending SCM replica requests, select the SCM replicas that you want to approve.
 - Click **Approve** to approve the replica.
 - Click **Reject** to reject the replica and remove it from the list.

When a replica is approved, it is listed in the **SCM Integrations** page beneath the master SVN server.

Related Links

*Convert to a replica server in TeamForge***Edit replica settings**

As a TeamForge site administrator, you can configure replica settings for the polling frequency of the master, and repository initialization and synchronization events.

Replication events, such as creating new repositories and synching commits, are stored in a queue on the TeamForge application server. The replica server polls the TeamForge application server for new events and then processes those events on the replica server.

These events are divided into two separate pools:

- New repository initializations - this includes creating the repository and performing the initial synchronization of the content.
- All other events

When existing repositories are selected for replication, this can take a long time. It could take many hours or even days to fully replicate an entire repository across a WAN. These big events are processed in their own thread pool, so that other repositories which are already synchronized don't have to wait in line for them to finish.

For each pool, you can define how many simultaneous events will be processed. The higher the number, the greater the potential load on both the TeamForge replica server and the Subversion master. However, this can also decrease the wait time for a given commit to appear on the replica server.

1. Click **Admin** in the CollabNet TeamForge navigation bar.
2. On the site administration navigation bar, click **Integrations**.
3. On the **SCM Integrations** page, click the name of the Subversion Edge replica you want to edit.

The **Edit System** page for the replica appears. Here's an example:

Subversion Edge Replica

Name: *	<input type="text" value="California"/>		
Description: *	<input style="height: 20px;" type="text" value="cu135 in Brisbane cloud"/>	Command polling interval: *	<input style="width: 40px;" type="text" value="10"/> seconds
		Maximum simultaneous new repository initializations: *	<input style="width: 40px;" type="text" value="1"/>
		Maximum simultaneous repository synchronizations: *	<input style="width: 40px;" type="text" value="2"/>
Hostname:	cu135.cloud.sp.collab.net		
Master Integration Server:	svn		
Managed By:	SVN Edge Sync (svnedge)		
Approval Status:	Approved		
Last Contact:	Thu Feb 10 10:56:23 PST 2011		

Recent Replication Command History

Status	Replica Command	Repository	Created Date	Last Contact
✓	Sync repo	/svnroot/dev	Thu Feb 10 10:46:04 PST 2011	Thu Feb 10 10:46:14 PST 2011
✓	Sync repo	/svnroot/dev	Thu Feb 10 09:05:22 PST 2011	Thu Feb 10 09:05:24 PST 2011
✓	Sync repo	/svnroot/dev	Thu Feb 10 09:03:49 PST 2011	Thu Feb 10 09:03:54 PST 2011
✓	Sync repo	/svnroot/dev	Thu Feb 10 09:01:56 PST 2011	Thu Feb 10 09:02:04 PST 2011
✓	Sync repo	/svnroot/dev	Thu Feb 10 08:54:43 PST 2011	Thu Feb 10 08:54:54 PST 2011

Done

4. Change the replica name or description if required.



Tip: Including the geographic location would help users select a nearby replica.

5. Set **Command polling interval** to define how frequently the replica polls the master looking for new events. The replica will process all new events when it polls.

The default value for this setting is 60 seconds, but it can range from 5 to 1000000 seconds.

6. Set **Maximum simultaneous new repository creations** to a low value.

New repository initializations can take a long time and generate a lot of load. So you wouldn't want to allow too many of them to run at once. This value can range from 1 to 100, but we suggest you keep it at 3 or less.

7. Set **Maximum simultaneous repository synchronizations** taking into account how many repositories you will be replicating and how many you think are likely to have commits occurring within the polling interval.

This value can range from 1 to 100. You may want to set this higher than the previous field, but we suggest you keep it at 10 or less. There's no reason to enter too high a number because you are merely specifying how many synchs can run at the exact same time -- and it never runs more than one per repository.

8. When you've made your changes, click **Save**.

Remove a replica

When you remove a replica from TeamForge, it is restored to a Subversion Edge server in standalone mode.

1. Click **Admin** in the CollabNet TeamForge navigation bar.
2. On the site administration navigation bar, click **Integrations**.
3. In the **Edit System** page for the replica, click **Delete**.




The replica is removed from TeamForge. The repositories that existed on the replica are deleted.

Replica status icons


The icons represent the health of a replica server, the state of a replicated repository, or the status of a replication command.




You can get more information related to a specific status by clicking on its icon. However, not all icons are links.

Replica server





Icon	Description
	There are no errors. The replica server has polled the master within 3 times the Command polling interval . For example, if the polling frequency is 10 seconds, TeamForge has a record of contact within the last 30 seconds.
	This could mean one of two things: <ul style="list-style-type: none"> • The server has an error in the last 30 commands, although the most recent command is successful. • The server has not polled the master within 3 times the Command polling interval, although it has within 30 times the polling interval.
	This could mean one of two things: <ul style="list-style-type: none"> • The most recent command is a failure. • The server has not polled the master within 30 times the Command polling interval.

Replicated repository

Icon	Description
	Replicated repository is in sync with the repository on the master.

Icon	Description
	Replicated repository is being added.
	Replicated repository is being deleted.
	Replicated repository is being synchronized using <code>svnsync</code> .

Replication command

Icon	Description
	The command is queued in the TeamForge server for the replica.
	Depending on the replica server queue, the command is being processed or scheduled to be processed.
	The execution of the command failed in the replica server. You will see a link to the server log that contains details about the error.
	The execution of the command succeeded in the replica server.

Replicate a repository

When a Subversion Edge replica has been successfully registered with a TeamForge SCM integration server, it is available to project administrators in projects using that server to house repositories. To replicate a Subversion repository, you need to add it to a replica server.

Before you can replicate a Subversion repository, an administrator must first add one or more replica servers. This involves converting a [Subversion Edge server](#), and then [approving the replica](#) in TeamForge.

1. Click **Source Code** in the project navigation bar.
2. In the list of project repositories, select the one you want to replicate and click **Edit**.

The **Edit Repository** page lists the available replica servers. Here's an example:

If you don't see any available replica server listed here, it may be because none were created for this Subversion server, or there are pending replicas which haven't yet been approved by a TeamForge administrator.

3. In the Available Replica Servers section, select a replica server and click **Replicate Repository**. You will see it in the list of replica servers hosting the repository.
 - To see details such as the hostname and the user managing the replica repository, and recent replication command history, click the **Status** icon for the repository.
 - For the Subversion checkout command, see the project's **Source Code** page.



Note: The TeamForge account used while setting up the replica determines what's available to be checked out from a replica repository. This account – the **Managed By** user in the **Repository Replica Details** page – could have been provided total access to the master repository, or restricted access using path-based permissions.

Related Links

[Convert to a replica server in TeamForge](#)

Monitoring a replica server

Proactive monitoring of the replica server occurs via email. The TeamForge web interface displays recent command history including the commands that are scheduled to run, the status of the ones actively running and the ones that failed. For current jobs, there are links to the runtime output.

Active monitoring

Configuring the mail server information in the Subversion Edge replica turns on active monitoring. For more information, see [Configure the mail server](#). The replication process sends emails when an event fails. For example, if `svnsync` is running and the network drops, the retries are exhausted or not turned on and the command fails. An email is sent to the address configured in Subversion Edge — this can be a group address or mailing list. The email contains the same information as the log. This includes the error information in the output of the `svnsync` command as well as an internal stack trace.

Passive monitoring

The TeamForge web interface provides monitoring information. At the site level, you can see the status of the entire replica server and queue including the event actively being processed. You can also see this at the repository level from within a TeamForge project. For more information, see [Check command history in TeamForge](#).

There are basic summary icons to show status. For more information, see [Replica status icons](#). A cloud icon indicates a problem in the recent events and you can click it to see the details.

Usage monitoring

As with Subversion on a master server, the way to measure the usage of the replica server is via the logging. The replicas have both Apache access logging and Subversion command logging. You can create usage reports from these log files to find out how much activity is being serviced by the replica server.

Related Links

[Manage replicas in TeamForge](#)

Check command history in TeamForge

Recent command history for a replica server or a specific repository allows you to check for errors and see whether there are pending commands.

At the site level, you can see the status of the entire replica server and event queue including the event being actively processed. You can also see this at the repository level from within a TeamForge project. >

If a repository revision is not showing up, for example, you can check for errors to know it's not just because the repository is still busy synching the latest revision. You can also check whether there are commands waiting in the queue and whether the repository is truly in synch.

- To check the command history for a replica repository, follow these steps:
 - a) Click **Source Code** in the project navigation bar.
 - b) In the list of repositories, click the icon for the one you're interested in.
You'll see repository details and command history. Here's an example:

Repository Replica Details

Replica Server: Tokyo Hostname: cu126.cloud.sp.collab.net Managed By: Subversion Replication Last Contact: Wed Apr 06 12:23:32 PDT 2011	Server: Subversion Repository Name: Benchmark Description: SVN benchmarks
---	--

Replication Command History Page 1 of 3 (45 Items)

Status	Command	Revision	Created Date	Last Contact
Any			From 04/05/2011 (MM/dd/yyyy) To 04/06/2011 (MM/dd/yyyy)	
✓	Sync repository	70	Wed Apr 06 12:21:50 PDT 2011	Wed Apr 06 12:21:54 PDT 2011
✓	Sync repository	69	Wed Apr 06 12:21:46 PDT 2011	Wed Apr 06 12:21:54 PDT 2011
✓	Sync repository	68	Wed Apr 06 12:21:41 PDT 2011	Wed Apr 06 12:21:43 PDT 2011



Tip: You can also check command history by clicking on the **Status** icon in the **Edit repository** page.

- You need to be a TeamForge administrator to check the command history across a replica server.
 - a) On the site administration navigation bar, click **Integrations**.
 - b) On the **SCM Integrations** page, click the name of the replica server you're interested in. The **Edit System** page displays the command history.

From within the event list, you can click on a failure to be taken to the Subversion Edge console and the log file with the details of the problem.

You can also monitor current activity. For example, suppose there is a new repository initialization that has been syncing for several hours and you want to get an idea of how far it has progressed. You can click the event to be taken to the Subversion Edge server where the output of the `svnsync` process is tailed. You can then see what `svnsync` is doing and get an idea of what revision it is processing.

Related Links

[Convert to a replica server in TeamForge](#)

Update TeamForge account for replica mode

The TeamForge credentials you supply during conversion to replica mode are used for registration and communication with the core (master) Subversion server. When the conversion is complete, you can edit this account in the Subversion Edge console.

1. Select **Administration** in the main navigation bar and click **TeamForge Mode**.
2. Click the **Update TeamForge Credentials** tab.
3. Change the user name and password for your TeamForge account and click **Continue**.

Updates to TeamForge site administrator credentials are accepted here.

4. Provide the value of the [Integration API Key](#) configured on the TeamForge server.

You will see a message that your credentials have been updated.

Related Links

[Convert to a replica server in TeamForge](#)

Update replica retry settings

You can configure the wait time and number of retry attempts for replica commands that fail.

If you have locations with frequent network issues, this feature can be very useful because the replica will keep trying the same command with a delay period between each retry. If the network problem is resolved during the interval, the command will go through as if nothing happened. However, there is a trade-off in how long you want the processing of other commands to be blocked while one command retries and waits.

If the problem is one that is not going to resolve itself, such as `svnsync` not being able to handle the revision for some reason, or there is an RBAC (role-based access control) setup problem in TeamForge, this feature would waste time and block other transactions from happening. So you have to factor that in when configuring the number of retries and the delay.

1. Select **Administration** in the main navigation bar and click **TeamForge Mode**.
2. Click the **Update Replica Configuration** tab.
3. Specify how many times you want a failed replica command such as a repository synchronization to be tried.
4. Specify the wait time (in seconds) before a failed command is retried.

While waiting for a command or retrying it, other commands are blocked.

When a command has been retried the specified number of times and still does not succeed, it is marked as failed in TeamForge. At this point, an email is sent to the **Replica Sync Notification Address** (if you configured this while converting your Subversion Edge server to replica mode).

Convert to standalone mode

If you've converted Subversion Edge to a primary integration server or replica server in TeamForge, it is possible to revert to the standalone management console mode.

1. Select **Administration** in the main navigation bar and click **TeamForge Mode**.



Note: You'll see the **TeamForge Mode** link after a successful conversion to TeamForge.

2. On the **Convert to Standalone mode** page, supply your TeamForge administrator credentials.

This account must have permissions to remove integration servers in TeamForge.

3. Click **Convert**.

After the conversion, local authentication is used and console functionality is fully restored.








The console is put back as it was before you converted to TeamForge mode. Users and server settings that you had before the conversion still exist. Repository hook scripts are restored.

When you revert from a core TeamForge SCM integration server to standalone mode, what you see with repositories is based on what happened while you used TeamForge. If you created new repositories or deleted repositories, that's what you have after you convert to standalone mode. Repository content itself is not modified -- it is not rolled back to what it looked like before you converted to TeamForge mode.

When you revert from replica to standalone mode, you will once again be able to access the repositories you had before the conversion.

Status icons for replica jobs

These icons represent the status of the jobs running on the replica.

Icon	Description
	Copying revision properties of the repository.
	Converting the server to replica mode.
	Updating the replica server's properties.
	Converting the server back to console mode.
	Replicating master repository.
	Removing replicated repository.
	Synchronizing replicated repository.

Extend Subversion Edge with the CollabNet cloud

CollabNet Cloud Services offers free backup and recovery for your Subversion repositories. When you're ready to do more with the cloud, you can create projects, migrate your repositories and manage them with CloudForge.

Related Links

[CloudForge](#)

[CloudForge Support](#)

What is a cloud backup?

When you back up a Subversion repository to the CollabNet Cloud, a repository is created in the cloud. The `svnsync` tool is used to set up and push changes into the repository. The repository is periodically synched based on the backup schedule you set up.

Related Links

[Schedule a cloud backup](#)

[Access a cloud backup](#)

[Restore a repository from a cloud backup](#)

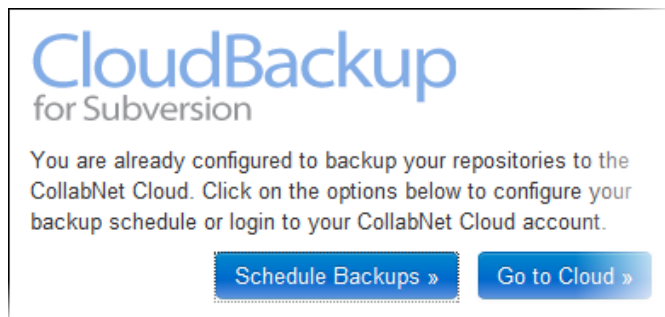
[svnsync](#)

Schedule a cloud backup

Protect your code by backing up your repositories to the CollabNet Cloud.

You need an account with CollabNet Cloud Services; when you register, you get 250 MB of free storage.

1. Select **Administration** in the main navigation bar and click **Cloud Services**.
2. Click **Schedule Backups**.



3. In the **Backup Schedule** page, select **Cloud Services Backup**.
4. In the **New Jobs** tab, select one or more repositories to back up.
5. Specify how often and at what times you want the repositories backed up.
6. Click **Create**.

You can check the status of cloud backup jobs in the **Administration > Jobs** page.

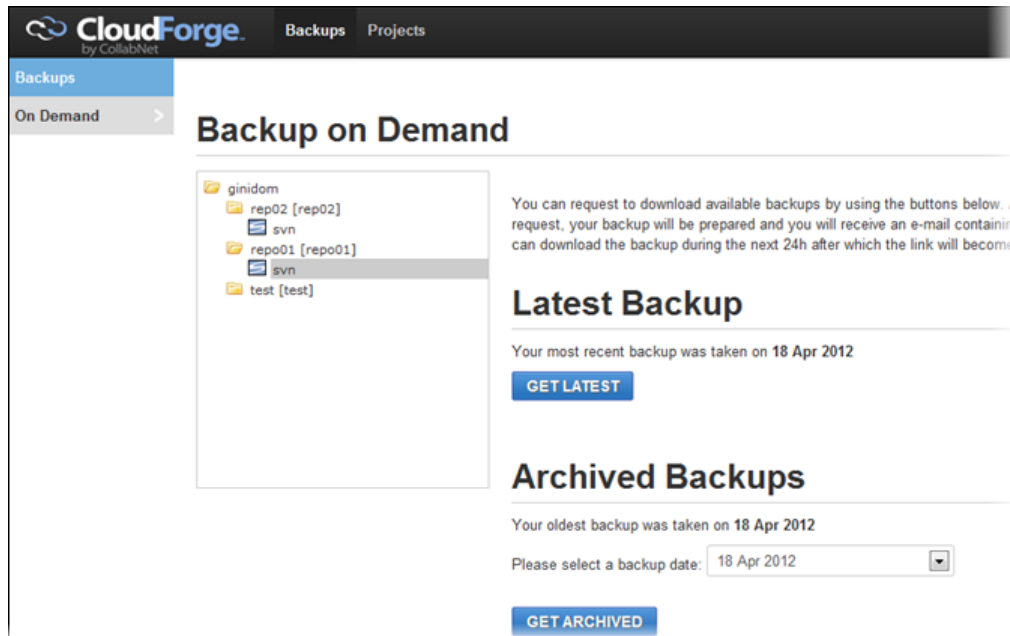
Related Links

[What is a cloud backup?](#)

Access a cloud backup

Log in using your Cloud Services account and request the backup you want. You will receive an email with the URL to download the backup.

1. Select **Administration** in the main navigation bar and click **Cloud Services**.
2. Click **Go to Cloud**.
3. Log in using your Cloud Services account.
4. In the CloudForge **Backup on Demand** page, select a Subversion repository and choose whether you want to get the latest backup or an archive.



When the backup is ready, you will receive an email containing a URL to download it.

Related Links

[What is a cloud backup?](#)

Restore a repository from a cloud backup

Let's say you lose your Subversion Edge server's hard drive. You can restore your repositories by creating new ones with the content from the cloud backups.

Here's what you'd need to do:

- [Install Subversion Edge and configure](#) the server.
- Use your existing credentials to connect to your CollabNet Cloud Services account.
- [Create a new repository](#) using the **Cloud Repositories** option.

Create Repository

Name:

Initialize: Template
 Backup
 Cloud Repositories

Select a repository from the cloud to restore as a new local repository.

rep02
 repo01

The repository is created on your local server and synchronized from the cloud backup.

You can check the status of this job in the **Administration > Jobs** page.

Related Links

[What is a cloud backup?](#)

Maintain your Subversion Edge site

You maintain your Subversion Edge site by monitoring and logging its performance.

Monitor server health

Use statistical data to gauge overall activity and monitor the health of the server.

- Select **Administration** in the main navigation bar and click **Status**.

Information such as how long the server has been running, disk space used by the repositories, available disk space, and network throughput is displayed.



Note: If you have the `ROLE_ADMIN` or `ROLE_ADMIN_SYSTEM` role, you also have an option to start or stop the server.

- To see server disk space and network utilization trends, select **Administration** in the main navigation bar and click **Statistics**.

You could use the data to determine how much the server is being used, as well as spot potential problems where hardware upgrades might be needed. The trend-based charts should assist with projecting future needs. For example, you could use the disk space charts to see the growth trends in repository disk space and project future disk needs.



Note: To see the disk space usage for an individual repository, select **Repositories** in the main navigation bar, click the status link for the repository and look at the **Size Report** tab.

Related Links

[Monitor the disk space used by a repository](#)

Monitor server logs

Use logs to debug problems and ensure that the application is performing to expectations.

You need the `ROLE_ADMIN` or `ROLE_ADMIN_SYSTEM` role to be able to view server logs.

1. Select **Administration** in the main navigation bar and click **Server Logs**.

The logs are displayed in the **Available Files** page.

Logs are rotated daily. You'll see one of each of these per day, with the date stamp appended to the name:

- `access.log`: the standard Apache access log that logs all requests to the CollabNet Subversion Apache server.
- `error.log`: records Subversion Apache server errors. The verbosity of the messages recorded here depends on the **Subversion Server Log Level** setting.
- `console.log`: records management console operations. The verbosity of the messages recorded here depends on the **Console Log Level** setting.
- `jetty.log`: records requests processed by the Jetty server.
- `subversion.log`: records summaries of high-level Subversion operations.
- `replica_cmds.log`: records commands executed on the Subversion Edge server that's been converted to a replica server in TeamForge.



Note: For a failed command, the replica server in TeamForge has a link to the log that contains the related error.

Logs are retained for the number of days specified by the **Delete log files older than** server setting.

- Click a log file to view its raw content.

Related Links

[Configure log settings](#)

Monitor jobs

As an administrator, you may find it useful to check the **Jobs** page for the status of repository verify and backup jobs and replica server commands.

Select **Administration** in the main navigation bar and click **Jobs**.

In the Subversion Edge standalone mode and TeamForge integration server mode, you'll see the active, scheduled and completed backups. Here's an example:

Jobs					
Active Background Jobs					
#	Id	Description	Scheduled for	Started at	Finished at
1	Idle				
Finished Background Jobs					
#	Id	Description	Scheduled for	Started at	Finished at
1	RepoDump-argoadml-hotcopy-XT001700	Backup hotcopy of repository 'argoadml' View result.	04/17/2012 12:17:00 PDT	04/17/2012 12:17:00 PDT	04/17/2012 12:17:00 PDT
2	RepoDump-coalescent-hotcopy-XT001700	Backup hotcopy of repository 'coalescent' View result.	04/17/2012 12:17:00 PDT	04/17/2012 12:17:00 PDT	04/17/2012 12:17:00 PDT
3	RepoDump-eent-hotcopy-XT001700	Backup hotcopy of repository 'eent' View result.	04/17/2012 12:17:00 PDT	04/17/2012 12:17:00 PDT	04/17/2012 12:17:00 PDT
4	RepoDump-ampx-hotcopy-XT001700	Backup hotcopy of repository 'ampx' View result.	04/17/2012 12:17:00 PDT	04/17/2012 12:17:00 PDT	04/17/2012 12:17:00 PDT
5	RepoDump-acceptor-hotcopy-XT001700	Backup hotcopy of repository 'acceptor' View result.	04/17/2012 12:17:00 PDT	04/17/2012 12:17:00 PDT	04/17/2012 12:17:00 PDT
Scheduled Background Jobs					
10	records per page				Filter:
#	Id	Description	Scheduled for		
21	RepoDump-sdt-hotcopy-XT001700	Backup hotcopy of repository 'sdt'	04/18/2012 12:17:00 PDT		
22	RepoDump-subclipse-hotcopy-XT001700	Backup hotcopy of repository 'subclipse'	04/18/2012 12:17:00 PDT		
23	RepoDump-subissue-hotcopy-XT001700	Backup hotcopy of repository 'subissue'	04/18/2012 12:17:00 PDT		

In the TeamForge replica server mode, you'll see the commands that are queued and waiting to be executed as well as the ones actively running.

Related Links

[Convert to a replica server in TeamForge](#)

Subversion Edge FAQ

Related Links

[Frequently Asked Questions wiki page](#)

Can I use the web console with my custom Apache configuration?

Yes. If you have an existing Apache configuration, you can still keep that and use the CollabNet Subversion Edge console.

In most scenarios, you should be able to set up the server using the web interface. For any additional configuration, you can edit `httpd.conf` located in `/data/conf` in the directory where you installed Subversion Edge. If you have the `ROLE_ADMIN` or `ROLE_ADMIN_SYSTEM` role, you will be able to start and stop the server from the console.

When I try to browse repositories I am getting an Internal Server Error. How do I fix this?

When Subversion Edge is installed, it adds itself to PATH. You may need to reboot your system after the initial install, before the PATH change is visible to the Windows ViewVC service.

How do I migrate from my existing Subversion installation to Subversion Edge?

Migrating from your existing Subversion installation, whether CollabNet Subversion, open-source Subversion or some other Subversion server product, is fairly straightforward.

See [this wiki](#) for details.

Using Subversion with TeamForge

Common questions on using Subversion 1.7 with your TeamForge installations.

Can I use a Subversion 1.7.x client with my TeamForge 6.1.1 server?

Yes. Subversion guarantees compatibility between any 1.x client and 1.x server.

This means you can use a Subversion 1.7.x client with any version of TeamForge (or SourceForge Enterprise Edition). Given that the Subversion 1.7.x client brings significant performance improvements, CollabNet encourages all users to adopt the Subversion 1.7.x client as soon as convenient.

Can I upgrade the Subversion server installed by TeamForge to version 1.7.x?

Not for TeamForge 5.4 and 6.1; with TeamForge 6.1.1, this is possible.

Subversion is provided as part of your TeamForge installation and is provided by the CollabNet YUM repository. The version of the Subversion server is maintained at the 1.6.x level for existing versions of TeamForge.

Future releases of TeamForge are expected to update Subversion to the 1.7.x release. However, if you are using TeamForge 5.4 or later, you can connect a Subversion Edge server to TeamForge as an external SCM integration server; using Subversion Edge, you can install the latest version of Subversion.

Related Links

[Upgrade the TeamForge 6.1.1 Subversion server to version 1.7.x.](#)

Can I install Subversion 1.7.x on a Subversion Edge server set up as a TeamForge integration server?

Yes. Just log into the Subversion Edge server and click the button to install the latest updates.

In the Subversion Edge 2.1.0 release, Subversion 1.7.x binaries are included by default.

Are there any other upgrades I need to consider when upgrading to Subversion 1.7.x?

No. Just install the updates via the Subversion Edge updater and everything is taken care of automatically.

Upgrading to Subversion 1.7.x involves installing the latest binaries and restarting the Apache server. The Subversion Edge updater takes care of this for you.

Can I install Subversion 1.7.x on a Subversion Edge server set up as a TeamForge replica server?

Yes. Just install the latest updates to Subversion Edge.



Note: Until the master SVN server is running Subversion 1.7.x or later, the replica can only expose Subversion 1.6.x features to the clients. This mainly means that clients cannot take advantage of the new HTTPv2 protocol improvements until the master has been upgraded to 1.7.x too. Subversion Edge takes care of all the details -- there is nothing you need to do or configure.

Subversion Edge REST API

Starting with version 2.2, Subversion Edge provides a RESTful API. The API enables other applications, for example, the CollabNet Desktops, to connect to Subversion Edge. Simple, script-based interactions could also use the API endpoints to automate tasks such as migrating users.

The current version of the API is 1.0. Additions to the API will not cause the version to increment.

General usage

The API is accessible at the same location as your Subversion Edge console.

For example, if you access your console at `http://host.mycompany.com:3343/csvn`, the API URL would look like `http://host.mycompany.com:3343/csvn/api/1/repository/34`, where "1" indicates the API version number, "repository" indicates the entity you are working with, and "34" indicates the ID of the instance. In this documentation, such URLs are shown as absolute paths (`/csvn/api/1/repository` without the host, port number and so on).

Request format

Requests to the API are directed to a URL indicating the entity and use one of the following HTTP methods (verbs) to indicate the action:

- GET
- PUT
- POST
- DELETE

A request body is required for PUT or POST operations. It can contain XML or JSON data for parameters, or file content for certain posting operations such as uploading a zipped repository as a template. The Content-Type header indicates the format of the body. Response format is indicated with a `format` URL parameter, in the Accept header, or as a pseudo extension to the entity or instance (for example, `/csvn/api/1/repository/34.json`). Here is an example POST request in JSON and XML formats:

```
POST /csvn/api/1/repository?format=json
```

```
{
  "name": "new-repo",
  "applyStandardLayout": "false",
  "applyTemplateId": 2
}
```

```
POST /csvn/api/1/repository?format=xml
```

```
<map>
  <entry key="name">new-repo</entry>
  <entry key="applyStandardLayout">false</entry>
  <entry key="applyTemplateId">2</entry>
</map>
```

Authentication

The API requires basic HTTP authentication for all services except *Secure Port*. In a web browser, you can provide this authentication using the authentication dialog box or in the user-information section of the URL. You can also supply an Authorization header in the request, where the value for the header is "Basic {md5 hash of username:password}".

Response format

The API responds in XML or JSON format depending on the request type, and uses HTTP status codes to indicate success or failure. The most important status codes are as follows:

200	OK (for successful GET requests)
201	Entity created or updated successfully (for successful PUT, POST or DELETE)
400	Bad request (parameters incorrect or missing in the URL or request body)
401	Unauthorized
404	Not found
405	Method not implemented (for example, POST is not supported for this entity type)

Here is an example response in JSON and XML formats:

```
GET /csvn/api/1/repository?format=json
```

```
{
  "repositories": [
    { "id": 1,
      "name": "repo1",
      "status": "OK",
      "svnUrl": "http://apitest/svn/repo1",
      "viewvcUrl": "http://apitest/viewvc/repo1" },
    { "id": 2,
      "name": "repo2",
      "status": "OK",
      "svnUrl": "http://apitest/svn/repo2",
      "viewvcUrl": "http://apitest/viewvc/repo2" }
  ]
}
```

```
GET /csvn/api/1/repository?format=xml
```

```
<map>
  <entry key="repositories">
    <map>
      <entry key="id">1</entry>
      <entry key="name">repo1</entry>
      <entry key="status">OK</entry>
      <entry key="svnUrl">http://apitest/svn/repo1</entry>
      <entry key="viewvcUrl">http://apitest/viewvc/repo1</entry>
    </map>
    <map>
      <entry key="id">2</entry>
      <entry key="name">repo2</entry>
      <entry key="status">OK</entry>
      <entry key="svnUrl">http://apitest/svn/repo2</entry>
      <entry key="viewvcUrl">http://apitest/svn/repo2</entry>
    </map>
  </entry>
</map>
```

Related Links

[General Usage wiki](#)

Subversion Edge User API

Use this API to create Subversion Edge users.

This API is available with Subversion Edge 2.3.0 (and later). It supports only the POST method.



Note: The following examples show request bodies and responses in JSON format, but XML is equally valid.

POST

HTTP method and URL

```
POST /csvn/api/1/user?format=json
```

Example:

```
POST /csvn/api/1/user?format=json
{
  "username": "example1",
  "password": "mysecret",
  "fullName": "Example Uno",
  "emailAddress": "example1@collab.net"
}
```

Parameters

username

Required. The login name must be unique.

password

Required.

fullName

Required.

emailAddress

Required.

Authentication / Authorization

User administrators (ROLE_ADMIN_USERS) and the super user (ROLE_ADMIN)

Response

On success (201), the `userId` of the new account is returned. On failure (400), the nature of the problem is indicated in the `errorDetail` field.

Examples:

```
{
  "message": "Entity created",
  "userId": 1001
}
```

```
{
  "errorMessage": "Failed to create user",
  "errorDetail": "The username must be unique"
}
```

Related Links

[User API wiki](#)

Subversion Edge User Roles API

Use this API to list available administrative roles and edit role membership.

This API is available with Subversion Edge 2.3.0 (and later).



Note: The following examples show request bodies and responses in JSON format, but XML is equally valid.

GET

HTTP method and URL
GET /csvn/api/1/role?format=json

Authentication User administrators (ROLE_ADMIN_USERS) and the super user (ROLE_ADMIN)

/

Authorization

Response The GET request returns a list of roles.

Example:

```
{ "roles": [
  {
    "id": "2",
    "authority": "ROLE_ADMIN_REPOS",
    "description": "Permits administration of repositories (create, backup, etc)"
  },
  {
    "id": "3",
    "authority": "ROLE_ADMIN_USERS",
    "description": "Permits administration of users (create, remove, etc)"
  }
]}
```

PUT

HTTP method and URL
PUT /csvn/api/1/role/2?format=json

Example:

```
PUT /csvn/api/1/role/2?format=json

{
  "userId": 1001,
  "action": "add"
}
```

Parameters The role ID, as well as user ID are required – the role ID in the URL path, and the user ID in the request body as a parameter.

userId

Required.

action

Optional. You can set it to either "add" or "remove".

Authentication /
Authorization

User administrators (ROLE_ADMIN_USERS) and the super user (ROLE_ADMIN)

Related Links[User Roles API wiki](#)

Subversion Edge Repository API

Use this API to list and create Subversion Edge repositories.

This GET method is available with Subversion Edge 2.2.0 (and later). The POST method requires Subversion Edge 2.3.0 (or later).



Note: The following examples show request bodies and responses in JSON format, but XML is equally valid.

GET

HTTP method and URL

```
GET /csvn/api/1/repository?format=json
```

Authentication / Authorization

All Subversion Edge users (ROLE_USER)

Response

GET retrieves the list of repositories in the Subversion Edge instance, including ID, name, status, and access URLs for each.

Example:

```
GET /csvn/api/1/repository?format=json

{
  "repositories": [
    { "id": 1,
      "name": "repo1",
      "status": "OK",
      "svnUrl": "http://apitest/svn/repo1",
      "viewvcUrl": "http://apitest/viewvc/repo1" },
    { "id": 2,
      "name": "repo2",
      "status": "OK",
      "svnUrl": "http://apitest/svn/repo2",
      "viewvcUrl": "http://apitest/viewvc/repo2" }
  ]
}
```

POST

HTTP method and URL

```
POST /csvn/api/1/repository?format=json
```

Example:

```
POST /csvn/api/1/repository?format=json

{
  "name": "new-repo",
  "applyStandardLayout": "false",
  "applyTemplateId": 2
}
```

Parameters

applyStandardLayout (boolean)

When POSTing a new repository, use this parameter for a standard "trunk/tags/branches" layout.

applyTemplateId (integer)

When POSTING a new repository, use this parameter to *apply a repository template*.

Authentication /
Authorization

Repository administrators (ROLE_ADMIN_REPO) and the super user (ROLE_ADMIN)

Related Links

[Repository API wiki](#)

Subversion Edge Repository Template API

Use this API to get a list of available repository templates, post a new template, or update an existing template.

This API is available with Subversion Edge 2.3.0 (and later).



Note: The following examples show request bodies and responses in JSON format, but XML is equally valid.

GET fetches a list of templates. When *creating a repository*, you can apply a template by using a template ID from this list.

GET

HTTP method and
URL

```
GET /csvn/api/1/template?format=json&showInactive=true
```

Parameters

showInactive

Optional. Provide this parameter to request all templates, including inactive or hidden ones.

Authentication /
Authorization

Repository administrators (ROLE_ADMIN_REPO) and the super user (ROLE_ADMIN)

Response

GET retrieves the list of repository templates in the Subversion Edge instance, providing the ID, name, and description for each. The list of active templates is returned; to have hidden templates returned as well, set the `showInactive` parameter in the query string to true. If the `showInactive` parameter is set to true, the results indicate whether each template is active or not.

Example:

```
GET /csvn/api/1/template?format=json&showInactive=true

{"templates":[
  {
    "id":1,
    "name":"Empty repository",
    "active":"true"
  },
  {
    "id":2,
    "name":"Create standard trunk/branches/tags structure",
    "active":"false"
  }
]}
```

PUT

HTTP method and URL	PUT /csvn/api/1/template/2?format=json
Authentication / Authorization	Repository administrators (ROLE_ADMIN_REPO) and the super user (ROLE_ADMIN)
Response	PUT returns the template ID and name.

Example:

```
PUT /csvn/api/1/template/2?format=json

{
  "message": "Entity was successfully created or updated",
  "template": {
    "id": 2,
    "name": "My Existing Template"
  }
}
```

POST

When posting a new template, or updating an existing template, the request body is presumed to consist of a dump file (application/octet-stream) or a repository archive (application/zip). You can provide name and active query parameters for POST operations.

HTTP method and URL	POST /csvn/api/1/template?format=json&name=My%20New%20Template&active=true
Authentication / Authorization	Repository administrators (ROLE_ADMIN_REPO) and the super user (ROLE_ADMIN)
Response	POST returns the template ID and name.

Example:

```
POST /csvn/api/1/template?format=json&name=My%20New%20Template&active=true

{
  "message": "Entity was successfully created or updated",
  "template": {
    "id": 3,
    "name": "My New Template"
  }
}
```

Related Links

[Repository Template API wiki](#)

Subversion Edge Repository Hook Scripts API

Use this API to post a new hook script for a repository or replace an existing one.

This API is available with Subversion Edge 2.3.0 (and later).



Note: The following examples show request bodies and responses in JSON format, but XML is equally valid.

GET

HTTP
method and
URL

```
GET /csvn/api/1/hook/{repoId}
GET /csvn/api/1/hook/2?format=json&sort=date&order=desc
```

Parameters **scriptName**

Optional. When the URL path contains a script name, the response includes the file's contents. ASCII files have a "text/plain" mime-type. All other files, including non-ASCII encoded text files, are sent as "application/octet-stream".

Example:

```
GET /csvn/api/1/hook/{repoId}/{scriptName}
GET /csvn/api/1/hook/2/pre-commit
```

Authentication Repository hooks administrators (ROLE_ADMIN_HOOK) and the super user (ROLE_ADMIN)

/

Authorization

Response GET retrieves the list of repositories in the directory where the repository's hook scripts are located. Sub-directories are not included. The `repoId` field after the "hook" token refers to the repository.

Example:

```
GET /csvn/api/1/hook/2?format=json
{
  "hooks": [ {name: "pre-commit", size: 1024, lastModified: 134809548090}, {name: "post-commit", size: 398, lastModified: 134809548090} ]
}
```

PUT

Use the PUT method to create new hook scripts and update existing ones. The URL path provides the metadata about the binary or plain text file in the request body. The `repoId` field after the "hook" token refers to the repository in which the `scriptName` will be created or replaced by the request body content.

HTTP method and URL

```
PUT /csvn/api/1/hook/{repoId}/{scriptName}
PUT /csvn/api/1/hook/2/pre-commit?format=json
```

Authentication /
Authorization

Repository hooks administrators (ROLE_ADMIN_HOOK) and the super user (ROLE_ADMIN)

Response

The response indicates success or failure through the status code and message.

Example:

```
PUT /csvn/api/1/hook/2/pre-commit?format=json
{
  "message": "Entity was successfully created or updated"
}
```

DELETE

Use the DELETE method to remove files from a repository's hooks directory. Path parameters are the same as in the PUT method.

HTTP method and URL	DELETE /csvn/api/1/hook/{repoId}/{scriptName} DELETE /csvn/api/1/hook/2/pre-commit?format=json
Authentication / Authorization	Repository hooks administrators (ROLE_ADMIN_HOOK) and the super user (ROLE_ADMIN)
Response	The response indicates success or failure through the status code and message.

Example:

```
DELETE /csvn/api/1/hook/2/pre-commit?format=json

{
  "message": "Request completed successfully"
}
```

Related Links

[Repository Hook Scripts API wiki](#)

Subversion Edge Logging API

Use this API to view and configure logging for the Subversion Edge console and Apache server.

This API is available with Subversion Edge 2.3.0 (and later).



Note: The following examples show request bodies and responses in JSON format, but XML is equally valid.

GET

HTTP method and URL	GET /csvn/api/1/logging?format=json
Authentication / Authorization	System administrators (ROLE_ADMIN_SYSTEM) and the super user (ROLE_ADMIN)
Response	GET retrieves the log settings in the format accepted in the PUT request body.

Example:

```
{
  "consoleLogLevel": "DEBUG",
  "serverLogLevel": "ERROR",
  "daysToKeep": 20
}
```

PUT

HTTP method and URL `PUT /csvn/api/1/logging?format=json`

Example:

```
PUT /csvn/api/1/logging?format=json
{
  "consoleLogLevel": "DEBUG",
  "serverLogLevel": "ERROR",
  "daysToKeep": 20
}
```

Parameters

consoleLogLevel

This is the log level for the Subversion Edge console. You can set it to one of the following: DEBUG, INFO, WARN, ERROR.

serverLogLevel

This is the log level for the Apache server. You can set it to one of the following: DEBUG, INFO, WARN, ERROR.

daysToKeep

This specifies the number of days of rotated log files to keep. To keep all log files, specify a value of 0.

Authentication /
Authorization

System administrators (ROLE_ADMIN_SYSTEM) and the super user (ROLE_ADMIN)

Related Links

[Logging API wiki](#)

Subversion Edge Secure Port API

Use this API to find out the SSL port number for HTTPS communications.

The Subversion Edge console runs simultaneous listeners for plain HTTP and HTTPS communications. Out of the box, the ports for these are 3343 and 4434 respectively. However, as the administrator, you can customize them and disable plain HTTP. If plain HTTP is disabled, the HTTP listener redirects to the HTTPS listener.

The [Discovery Service](#) advertises the plain HTTP location via mDns (Bonjour); so you can easily find the URL for plain HTTP. If you need to use HTTPS for API communications, you must first access the Secure Port API via plain HTTP to find the SSL port.

This API is available with Subversion Edge 2.2.0 (and later). It supports only the GET method.



Note: The following examples show request bodies and responses in JSON format, but XML is equally valid.

GET

HTTP method and URL `GET /csvn/api/1/securePort?format=json`

Authentication / Authorization All Subversion Edge users (ROLE_USER)

Response

The response contains the SSL port number and whether or not SSL is required for all other API calls are indicated in the response.

Example:

```
{  
  "SSLPort": 4434,  
  "SSLRequired": true  
}
```

Related Links

[Secure Port API wiki](#)

[Discovery API wiki](#)

Reference

Supported software

This is the official list of software that is compatible with Subversion Edge.

Operating systems

Subversion Edge has been tested on these operating systems:

Operating system	Version	Architecture
Red Hat Enterprise Linux	5.x - 6.x	32-bit, 64-bit
CentOS	5.x - 6.x	32-bit, 64-bit
SuSE Linux Enterprise Server	11	32-bit, 64-bit
Solaris	10	32-bit, 64-bit
Windows XP		
Windows Vista		
Windows 7		
Windows 2003 Server (including R2)		32-bit, 64-bit
Windows 2008 Server (including R2)		32-bit, 64-bit



Note:

- Windows: The Windows distribution includes and installs the Python 2.5.4 runtime needed for proper operation of Subversion Edge.
- Linux: The Linux distribution relies on the version of Python distributed with the core operating system distribution. Subversion Edge includes the necessary components to support Python versions 2.4, 2.5 and 2.6 and should work on any Linux distribution that includes one of those Python versions. It does NOT work on distributions like Red Hat Enterprise Linux 4 which includes Python 2.3.
- Solaris: Subversion Edge requires Python 2.4 to 2.6. Unlike most Linux distributions, Solaris installations do not necessarily install or require Python to be installed. For our testing, we used the Python 2.4 version provided on the Solaris 10u9 DVD (SUNWPython). The Python package from sunfreeware.com did not work with Subversion Edge because it is compiled statically and lacks the shared libraries needed for Apache mod_python.

Java JRE/JDK

Subversion Edge requires a Java 1.6 JRE/JDK.

Related Links

[Supported Operating Systems wiki](#)

Subversion Edge components

CollabNet Subversion Edge is comprised of a number of components, with varying licenses.

See [this wiki](#) for the list of components.

Scripts

System administrators can use these utilities to control the behavior of the application.

csvn

The `csvn` script lets you control and monitor the Subversion Edge console application from the command line.

Usage

Example: To start the console, run this command from the directory where you installed Subversion Edge.

```
$ bin/csvn console
```



Tip: To see all the available options for the script, run `$ csvn`.

csvn-httpd

The `csvn-httpd` script lets you control and monitor the Subversion Edge Apache server from the command line.

Usage

Example: To start Apache server, run this command from the directory where you installed Subversion Edge.

```
$ bin/csvn-httpd start
```

Options

start

stop

status

install

remove

Subversion Edge Release Notes

CollabNet Subversion Edge 4.0.4

Release Date: December 2013

Features and enhancements

Subversion Edge 4.0.4 adds these new features and enhancements.

Highlights

Artifact ID	Description
artf9363	Subversion Edge 4.0.4 supports Subversion 1.8.5.

CollabNet Subversion Edge 4.0.3

Release Date: November 2013

Features and enhancements

Subversion Edge 4.0.3 adds these new features and enhancements.

Highlights

Artifact ID	Description
artf9273	Subversion Edge 4.0.3 supports Subversion 1.8.4.

Resolved issues

Subversion Edge 4.0.3 resolves these issues.

Artifact ID	Description
artf9196	The verification step is removed when replicating a repository that already exists.

CollabNet Subversion Edge 4.0.2

Release Date: September 2013

Features and enhancements

Subversion Edge 4.0.2 adds these new features and enhancements.

Highlights

Artifact ID	Description
artf9033	Subversion Edge 4.0.2 supports Subversion 1.8.3.

Resolved issues

Subversion Edge 4.0.2 resolves these issues.

Artifact ID	Description
artf9072	It was not possible to save a value of 0 for the LDAPConnectionPoolTTL setting.

CollabNet Subversion Edge 4.0.1

Release Date: August 2013

Features and enhancements

Subversion Edge 4.0.1 adds these new features and enhancements.

Highlights

Artifact ID	Description
artf8000	Subversion Edge 4.0.1 supports Subversion 1.8.1 and Apache 2.4.6.

Resolved issues

Subversion Edge 4.0.1 resolves these issues.

Artifact ID	Description
artf7930	Text on the Advanced Server Settings page contained the word "from" twice.
artf7952	HTML in input fields has been sanitized to protect against XSS vulnerabilities.
artf7964	Console login did not work for LDAP when the realm name was customized.
artf7985	When setting up a replica with Subversion Edge 4.0, a database error was logged.
artf7986	With Subversion Edge 4.0 replicas, the user interface did not prompt you to accept the SSL certificate of the master server. Consequently replications failed when SSL was being used.

Artifact ID	Description
artf7997	Code which caused cloud backups to fail as a result of failing to create a CloudForge session is now fixed.

CollabNet Subversion Edge 4.0.0

Release Date: June 2013

Features and enhancements

Subversion Edge 4.0.0 adds these new features and enhancements.

Highlights

Artifact ID	Description
artf7716	Apache HTTP Server has been upgraded from version 2.2 to 2.4. If you have manually customized the Apache configuration files or added third-party modules, you will now need to do this again. For more information on Apache 2.4.x, see http://httpd.apache.org/docs/2.4/new_features_2_4.html .
artf7717	Subversion Edge 4.0.0 supports <i>Subversion 1.8</i> .
artf5586 , artf7721 , artf7727	This release has a new page for advanced settings. While most uses will not need to modify these settings, this page makes it easy to control some of the settings that are defined in the configuration files without needing to manually customize the configuration files. For more information, see <i>Configure advanced server settings</i> .
artf7729	Subversion 1.8 checks the access rules before showing a list of repositories. If you do not have read access to a repository, you will no longer see it in the list. While the content of your repositories has always been secured by access rules in Subversion Edge, this new feature now means that the list of repositories displayed in the console respects these rules as well.
artf7730	You can now rotate logs by size in addition to date, and turn off the Apache access or Subversion logs. You also have an option to reduce some of the "noise" that will be created by new Subversion 1.8 clients. For more information, see <i>Configure log settings</i> .

Other enhancements

Artifact ID	Description
artf5009	There are improvements to how ViewVC help is displayed.
artf7757	This release incorporates a recent change in TeamForge to update commit objects when svn:log is updated.
artf7786	Configuration files are generated from templates in the <code>dist/</code> folder.
artf7787	ViewVC has been upgraded to version 1.1.20.
artf7796	SVNPathAuthz <code>short_circuit</code> is supported in TeamForge replica mode.
artf7907	The default port used by Subversion Edge for logging in to the console via LDAP has been changed to avoid a port conflict with the <code>wininit.exe</code> service.
artf7919	If your <code>httpd.conf</code> file was customized, you will see a warning message on upgrading to Subversion Edge 4.0.0. The file you customized will be archived.

Resolved issues

Subversion Edge 4.0.0 resolves these issues.

Artifact ID	Description
artf7782	The hotcopy archive of a symlinked repository directory had a different top-level directory than other hotcopy archives.
artf7802	The layout on the login page was broken when the browser was set to Portuguese.
artf7917	An incorrect Apache warning that appeared when you used port 443 and SSL is no longer displayed.

CollabNet Subversion Edge 3.3.2

Release Date: June 2013

Features and enhancements

Subversion Edge 3.3.2 includes Subversion 1.7.10 binaries.

Known Issues

The following issue is known to exist in CollabNet Subversion Edge 3.3.2.

This issue existed in previous releases as well, but the improved SSL support in this release makes it more likely you will run into this. When both the Subversion Edge console and the Apache server are configured for SSL, the first time you click on a link to the Apache server from the console the browser does not connect to the server. This mainly seems to happen with Internet Explorer, although it might also happen with Chrome on Windows.

Clicking the link again works, as do direct bookmarks to the URL.

CollabNet Subversion Edge 3.3.1

Release Date: April 2013

Resolved issues

Subversion Edge 3.3.1 resolves these issues.

Artifact ID	Description
artf7773	Syntax highlighting in ViewVC was broken.
artf7774	ViewVC character encoding detection is enabled by default.
artf7779	An error related to LDAP user email is fixed.
artf7781	When saving the user's input value for the repository parent path, the symbolic link is no longer converted.

Known Issues

The following issue is known to exist in CollabNet Subversion Edge 3.3.1.

This issue existed in previous releases as well, but the improved SSL support in this release makes it more likely you will run into this. When both the Subversion Edge console and the Apache server are configured for SSL, the first time you click on a link to the Apache server from the console the browser does not connect to the server. This mainly seems to happen with Internet Explorer, although it might also happen with Chrome on Windows.

Clicking the link again works, as do direct bookmarks to the URL.

CollabNet Subversion Edge 3.3.0

Release Date: April 2013

Features and enhancements

Subversion Edge 3.3.0 adds these new features and enhancements.

Artifact ID	Description
artf7194	Apache log rotation has been changed to use the <code>rotatelogs</code> program.
artf7660	In general, tips have been reviewed and updated.
artf7661	Subversion Edge 3.3.0 has a new Getting Started wizard.
artf7669	The server configuration for SSL has been modified to protect against BEAST attacks.
artf7709	Default Subversion Edge run levels have been changed to support NFS storage.
artf7714	Bootstrap has been updated to version 2.3.1.
artf7731	When the folder used to store repositories is owned by <code>root</code> , a message with an example command for changing the owner is displayed.
artf7733	Jetty has been upgraded to version 8.1.9.v20130131. For more information, see https://ctf.open.collab.net/sf/wiki/do/viewPage/projects.svnedge/wiki/Jetty_8_Upgrade .
artf7747	ViewVC has been upgraded to version 1.1.18.
artf7749	Email for LDAP users has been improved.
artf7756	Daemon scripts on Unix have been improved.
artf7758	Subversion Edge 3.3.0 supports Subversion 1.7.9.

Resolved issues

Subversion Edge 3.3.0 resolves these issues.

Artifact ID	Description
artf7197	Repository verification jobs that fail are indicated by a change in the status field.
artf7265	The default value for Administrator Email caused an error when mail configuration was tested.
artf7461	Zip files greater than 4 GB are handled better.

Artifact ID	Description
artf7587	When scheduling cloud backups for repositories containing the "-" (dash) character in their names, an error message appears indicating that this is an invalid character.
artf7588	Clicking the Replica Setup tab from the Convert to Replica Mode tab caused a "Null Pointer Exception".
artf7589	The Server Port field now rejects non-numeric characters and has an upper limit of 65535.
artf7662	The usage of <code>mod_python</code> with ViewVC has been removed.
artf7668	When log files were viewed in the web interface, leading spaces appeared in the first line.
artf7684	Invalid Subversion checkout URLs were displayed on the Repository List page for Subversion Edge in the TeamForge integration server and replica modes.
artf7688	The PID file handled by <code>root</code> is handled better.
artf7708	Unused <code>jar</code> files have been removed.
artf7732	Since the TeamForge integration now uses the C-based <code>authz</code> , <code>ScmPermissionsProxyServlet</code> has been removed.
artf7734	The code for <code>SessionProgressListener</code> has been updated to clarify that the session exists already.
artf7739	It was not possible to save repository access rules when the <code>svn_access_file</code> size was large. This has now been fixed by increasing the <code>MaxFormSize</code> setting in Jetty from the default of 200 KB to 10 MB. See these instructions for more information on how you can configure this setting yourself.
artf7742	The <code>appserver/work</code> folder was missing on Windows.
artf7753	When the server was configured for SSL on port 443 and LDAP authentication, it did not start.

CollabNet Subversion Edge 3.2.2

Release Date: December 2012

Features and enhancements

Subversion Edge 3.2.2 adds these new features and enhancements.

Artifact ID	Description
artf7670	Subversion Edge 3.2.2 supports Subversion 1.7.8.

Resolved issues

Subversion Edge 3.2.2 resolves these issues.

Artifact ID	Description
artf7666	OpenSSL has been built without <code>zlib</code> support to protect against CRIME attacks.
artf7667	The <code>svnsync init</code> command is now executed on successive attempts to add a repository to a replica.
artf7575	The <code>csvn-httpd</code> script has been improved to support the <code>http_bind</code> option.

CollabNet Subversion Edge 3.2.1

Release Date: November 2012

Resolved issues

Subversion Edge 3.2.1 resolves these issues.

Artifact ID	Description
artf7648	With Apache 2.2.23, there were LDAP authentication failures on Windows.
artf7649	An initial replication of one repository blocked sync to other existing replicated repositories.

CollabNet Subversion Edge 3.2

Release Date: October 2012

Features and enhancements

Subversion Edge 3.2 adds these new features and enhancements.

Artifact ID	Description
artf6106	There is an internal mechanism that skips the TeamForge integration wizard and just bootstraps an install into TeamForge mode.
artf6287	Some Hibernate warnings in the logs have been resolved.
artf7195	Subversion Edge 3.2 provides a new interface for <i>configuring system monitoring</i> .
artf7226	The sign-up form for CollabNet Cloud Services is populated based on information from the user logged in.
artf7441	When users are notified on new updates, there is a link to a wiki page with more information about the release.
artf7553	Email notifications are sent when replica commands fail. For more information on the new field for configuring this email address, please see " <i>Convert to a replica server in TeamForge</i> ".
artf7566	With Subversion Edge 3.2, you can <i>specify the wait time and retry attempts for replica commands that fail</i> .
artf7567	The Subversion Edge 3.2 user interface lets you manually replay replica commands.
artf7570	This release supports Apache 2.2.23.
artf7577	Subversion Edge 3.2 binaries have been updated to Subversion 1.7.7.
artf7592	ViewVC has been upgraded to version 1.1.16.

Resolved issues

Subversion Edge 3.2 resolves these issues.

Artifact ID	Description
artf7061	A ViewVC title bar bug has been resolved.
artf7281	When a new user is created, there is a field for confirming the password.
artf7495	Hotcopy and dump backups are now considered separately when removing older backups.
artf7564	Converting to TeamForge mode did not create user accounts. This is now fixed.
artf7578	A problem with converting to replica mode is now fixed.
artf7579	A problem with setting up cloud backups is now fixed.

CollabNet Subversion Edge 3.1.2

Release Date: September 2012

Resolved issues

Subversion Edge 3.1.2 resolves these issues.

Artifact ID	Description
artf7510	On the Scheduled Jobs page, an exception was thrown when you tried to create a Verify Repository job.
artf7534	On a Unix 64-bit system with Python version greater than 2.4, the TeamForge authorizer module did not load.

CollabNet Subversion Edge 3.1.1

Release Date: August 2012

Added features

Subversion Edge 3.1.1 adds these new features.

Artifact ID	Description
artf7481	Subversion Edge 3.1.1 supports Subversion 1.7.6.

Resolved issues

Subversion Edge 3.1.1 resolves these issues.

Artifact ID	Description
artf7254	The Cloud API has been updated to api.cloudforge.com.
artf7407	Dump files were saved as compressed files when the Compress option was not selected.

Artifact ID	Description
artf7442	The LDAP Base DN value is not mandatory. If no value is provided, the correct LDAPURL value, with a "/" that was missing in earlier releases, is written in the configuration.
artf7456	Subversion Edge 3.1.1 is more resilient about SIGAR errors. Errors in loading the SIGAR library do not block users from the running the application.
artf7459	When a backup job fails, Java exceptions are now logged.

CollabNet Subversion Edge 3.1.0

Release Date: July 2012

Added features

Subversion Edge 3.1.0 adds these new features.

Artifact ID	Description
artf7196	ViewVC has been updated to version 1.1.15.
artf7253	Subversion Edge 3.1.0 supports TeamForge 6.2.

Resolved issues

Subversion Edge 3.1.0 resolves these issues.

Artifact ID	Description
artf7273	The Administration > Jobs page now uses a 24-hour clock.
artf7275	Backup jobs were failing because the <code>logs/temp</code> folder was missing.
artf7296	There were problems with deleting users with <code>ROLE_ADMIN</code> permissions.
artf7404	For several users, it was not possible to send mail when the server did not require authentication.

CollabNet Subversion Edge 3.0.1

Release Date: May 2012

Added features

Subversion Edge 3.0.1 adds these new features.

Artifact ID	Description
artf7258	Subversion Edge 3.0.1 supports Subversion 1.7.5.

Resolved issues

Subversion Edge 3.0.1 resolves these issues.

Artifact ID	Description
artf7230	Cloud backup failure emails now include better details.
artf7231	A failed cloud backup job left behind progress files that caused subsequent backup jobs to be skipped.
artf7236	Some backup jobs created with versions prior to Subversion Edge 3.0.0 could not be deleted.
artf7255	When a repository is deleted, all backup jobs scheduled for the repository are now deleted.
artf7257	When repository names contained JSON characters such as an apostrophe, Subversion Edge produced invalid JSON and caused the page to not render.
artf7259	Subversion Edge web pages now include a meta tag to have Internet Explorer render them correctly.

CollabNet Subversion Edge 3.0.0

Release Date: April 2012

Added features

Subversion Edge 3.0.0 adds these new features.

Artifact ID	Description
artf5082	Apache status detection has been improved.
artf5084	The user interface for repositories and users has been improved using of jQuery DataTables.
artf6286	You can now filter the list of repositories.
artf7028	The new user interface for Subversion Edge uses the latest HTML5, CSS3 and JavaScript standards.
artf7029	The look and feel of ViewVC templates has been updated.
artf7030	The Status page is now available under Administration .
artf7032	The left navigation pane has a new tips component that contains context-sensitive tips related to Subversion, Subversion Edge and other CollabNet products and services.
artf7033	You can now schedule multiple backups for a repository.
artf7038	When TeamForge site administrators log in to Subversion Edge, they can see the repositories.
artf7057	Subversion Edge 3.0.0 unveils the <i>cloud backup</i> feature.
artf7063	Tables have been converted to data tables to allow for a dynamic number of items/page, searching, sorting, and more.
artf7064	Backups are only performed for repositories that have been changed since the last backup.

Artifact ID	Description
artf7076	Subversion Edge 3.0.0 supports Subversion 1.7.4 binaries.
artf7127	Jobs are now displayed in TeamForge mode.
artf7176	The performance of Discover Repositories has been improved.
artf7177	There is a new job to <i>verify repositories</i> .
artf7178	Unnecessary columns have been removed from the Scheduled Jobs table..
artf7210	UNC pathnames are supported on Windows.
artf7216	For binary files, ViewVC provides a link to download the file instead of displaying file content.

Resolved issues

Subversion Edge 3.0.0 resolves these issues.

Artifact ID	Description
artf7003	Emails are no longer sent to the default admin@example.com and devnull@collab.net accounts.
artf7058	The log viewer resizes to fit the browser when the browser is resized.
artf7059	Sort order for log files did not retain the last setting.
artf7060	The option to start the Subversion server automatically has been removed.
artf7065	The number of active backup jobs running concurrently is now limited to 3.
artf7080	Hotcopy backups failed when you had symlinks.
artf7098	A broken link to replication help has been fixed.
artf7209	Often, the launch page for Subversion Edge continuously showed the server as not being ready.

CollabNet Subversion Edge 2.3.0

Release Date: February 2012

Added features

Subversion Edge 2.3.0 adds these new features.

Artifact ID	Description
artf6353	The Convert Usernames to Lower Case option in the Authentication tab allows for access rules to be case-insensitive.
artf6598	You can <i>set up the details of your corporate email server</i> in the Subversion Edge console. When a scheduled background job encounters a problem, an email with information from the error is sent to different accounts depending on the situation.
artf6890	API documentation is available in <i>this wiki</i> .
artf6891	You can <i>manage repository hook scripts</i> with create, copy, rename, edit, download and delete options.

Artifact ID	Description
artf6989	Subversion Edge 2.3.0 supports Apache 2.2.22.
artf6990	This release supports Subversion 1.7.3.

Resolved issues

Subversion Edge 2.3.0 resolves these issues.

Artifact ID	Description
artf6320	Only one user at a time is allowed to edit the access rules file.
artf6987	A locking problem with editing access rules is fixed.
artf6991	Software update dialogs now work with Internet Explorer 9.
artf7050	While editing replica credentials, an error message is displayed if the API key is invalid.

CollabNet Subversion Edge 2.2.1

Release Date: January 2012

Resolved issues

Subversion Edge 2.2.1 resolves these issues.

Artifact ID	Description
artf6872	When a proxy was configured, reloading software updates caused an error.
artf6873	The Software Updates page no longer displays proxy details; instead, it now links to the Proxy Settings page if there is one configured.
artf6875	The HTTP TRACE method is disabled in Apache and Jetty. The TraceEnable Off directive has been added to <code>httpd.conf.dist</code> ; this will only affect new installs unless <code>data/conf/httpd.conf</code> is deleted. For existing installations, you can either add the directive to <code>httpd.conf</code> or delete the file so that it can be recreated from the default file included with Subversion Edge.
artf6879	Repository backup jobs failed when the size of the zip file was greater than 4GB.
artf6885	On the Proxy Settings page, the Port Number was formatted incorrectly.
artf6898	The SSL certificates for Codesion have been updated.
artf6968	An error occurred on trying to access ViewVC in Subversion Edge after conversion to TeamForge mode against TeamForge 5.4.
artf6971	Checkouts did not work with a Subversion Edge 2.2 replica that used a proxy to connect to TeamForge.

CollabNet Subversion Edge 2.2.0

Release Date: December 2011

Added features

Subversion Edge 2.2.0 adds these new features.

Artifact ID	Description
artf6441	In this release, mod_python is upgraded to version 3.3.1.
artf6596	Subversion Edge 2.2 includes API methods to <i>list repositories</i> and return <i>SSL access information</i> .
artf6616	Subversion Edge administrators can <i>create templates</i> from repository dump files or zip archives and use them to initialize new repositories.
artf6617	When you <i>create a new repository</i> , you can initialize it with an existing backup.
artf6796	This release provides improved <i>support for proxy servers</i> .
artf6842	Subversion Edge 2.2 supports Java 7.
artf6843	Subversion Edge 2.2 supports Subversion 1.7.2.

Resolved issues

Subversion Edge 2.2.0 resolves these issues.

Artifact ID	Description
artf6229	There was a problem with the upload command result for an expired session; this affected the ability to replicate large repositories.
artf6439	404 errors for "alert.css" occurred on many pages when watching network calls in Firebug (or the like).
artf6592	A confirmation message is displayed when a repository is created successfully.
artf6593	In TeamForge integration server mode, the jobs tab is now hidden.
artf6594	In TeamForge integration server mode, backup jobs do not run.
artf6595	A problem related to the protocol in an LDAP server's hostname is fixed.
artf6614	Some problems with capturing network statistics on Windows are fixed.
artf6682	A problem with the Jobs page not displaying job executions is fixed.
artf6704	Logging for skipped repository dump jobs has been improved.
artf6733	When converting to TeamForge mode, the TeamForge Credentials page asks for the Integration API Key.
artf6815	Some problems with handling replication initialization errors are fixed.
artf6834	There was a problem with updating TeamForge credentials for a replica. Now the current replica user's password may be updated, if it has changed in TeamForge. However, to change the user associated with the replica, the current user's credentials must allow login to TeamForge. An appropriate message is displayed if that is not the case.

Artifact ID	Description
artf6835	You get a more informative error message when the when TeamForge server is not reachable in integration server mode or replica server mode.
artf6844	Python bindings were not working correctly on Linux and Solaris.
artf6845	Windows installers are signed with the CollabNet certificate.
artf6846	For users who do not have Java installed, Java 6u29 is shipped with the Windows installer.

CollabNet Subversion Edge 2.1.1

Release Date: November 2011

Added features

Subversion Edge 2.1.1 adds these new features.

Artifact ID	Description
artf6683	Subversion Edge 2.1.1 supports Subversion 1.7.1.

Resolved issues

Subversion Edge 2.1.1 resolves these issues.

Artifact ID	Description
artf6608	For a Subversion Edge 2.0.1 server that was converted to replica mode, the upgrade to version 2.1.1 failed when the replica did not have any replicated repositories. This is now fixed.
artf6624	The exception that caused a server's failure to restart after upgrading to Subversion Edge 2.1.0 on Linux is now fixed.
artf6655	When the API key was invalid, converting a Subversion Edge server to an integration server in TeamForge appeared successful, but did not actually work. This is now fixed.
artf6681	Re-adding a repository to a replica when the first initialization was forced to stop prematurely for some reason, failed. This is now fixed.

CollabNet Subversion Edge 2.1.0

Release Date: September 2011

Added features

Subversion Edge 2.1.0 adds these new features.

Artifact ID	Description
artf4998	Subversion Edge 2.1.0 enables you to <i>load a repository from a dump file</i> .
artf4999	You can <i>perform a repository dump on demand</i> .
artf5001	With this release, you can <i>delete a repository</i> .

Artifact ID	Description
artf5997	Subversion Edge 2.1.0 supports Subversion 1.7.
artf6102	The instructions for using ports other than 80 or 443 have been updated.
artf6104	The size of the viewport for logs has been improved.
artf6205	You can now <i>schedule backups</i> of one or more repositories.
artf6220	In addition to Subversion Edge site administrators, the following users are now treated as replica administrators: <ul style="list-style-type: none"> • A TeamForge user with a global role for SCM Integrations permissions • The TeamForge user who configured the replica
artf6240	The openCollabNet tab is now called Extensions.
artf6429	Subversion Edge 2.1.0 supports Apache 2.2.21.
artf6415	There is a Hotcopy option for repository backups.

Resolved issues

Subversion Edge 2.1.0 resolves these issues.

Artifact ID	Description
artf5744	A user created with the ROLE_ADMIN option selected and the ROLE_USER option de-selected, can now access all tabs in the Subversion Edge console without encountering errors.
artf6074	Re-adding a replica after a failure to initialize a repository caused the repository to be out-of-date. The repository could be brought up to date only with a commit. This problem is now fixed.
artf6095	A HTML encoding issue on the Software Updates page is now fixed.
artf6165	An LDAP authentication bug that some users experienced after an upgrade to Subversion Edge 2.0.0 is fixed.
artf6211	The Windows service failed to install when the JAVA_HOME variable pointed to a location that did not exist. This is now fixed.
artf6319	<code>viewvc.conf</code> does not mention <code>httpd.conf</code> as a means of overriding it.

CollabNet Subversion Edge 2.0.1

Release Date: September 2011

Added features

Subversion Edge 2.0.1 adds these new features.

Artifact ID	Description
artf6230	This release includes improvements to the update mechanism in preparation for the 2.1.0 release. Installing updates on Windows from any official Subversion Edge release to a development build for version 2.0.1 or 2.1.0 caused the system to hang. This is now fixed.

CollabNet Subversion Edge 2.0.0

Release Date: June 2011

Added features

Subversion Edge 2.0.0 adds these new features.

Artifact ID	Description
artf5071	With Subversion Edge 2.0.0, CollabNet TeamForge 6.1 provides improved performance to globally distributed users by enabling replication of Subversion repositories to local sites. The replication is accomplished by a wizard-driven process in Subversion Edge. For more on replication, see the help .
artf5164	The Windows installer provides a way to apply new releases. For example, if you have Subversion Edge 1.3.x installed and running, you can run the Subversion Edge 2.0.0 installer to upgrade.
artf5347	Access rules are validated and errors, if any, are displayed.
artf5738	Subversion Edge 2.0.0 includes improvements to the Windows install of the service. As part of these changes, the installer will no longer update an existing JAVA_HOME variable; it will set the variable if it has not been set.
artf5892	Mixed case letters and periods are allowed in repository names.
artf5225	The Info link for a repository now shows many more details such as number of revisions, size, packed information, filesystem format, repository format, and sharding information.
artf5800	Integration server actions are logged.
artf5756	The LDAP Bind DN field has a warning and an example.
artf5914	Usernames starting with a number are allowed.

Resolved issues

Subversion Edge 2.0.0 resolves these issues.

Artifact ID	Description
artf5112	The Subversion Edge console allows you to delete an LDAP user.
artf5120	The JmDNS feature generated a flood of UDP packets. This is now fixed.
artf5162	In earlier versions, the Windows installer created a bunch of registry entries for Python. These entries are no longer needed, and are not created now.
artf5194	Python 2.7 is supported on Linux.
artf5735	Access rules containing non-ASCII characters did not work. This is now fixed and <code>svn_access_file</code> is encoded in UTF-8 format.
artf5204	The View Source option on the Authentication page no longer exposes the password for the LDAP server.
artf5205	The process for cleaning up log files after a certain number of days failed when that number was set to 30. This is now fixed.
artf5357	Restrictions on usernames (Login Name) have been relaxed. Usernames can include alphanumeric and special characters other than <code>)(\ \'\"&\"\$,<</code>

Artifact ID	Description
artf5755	A missing favicon.ico file, that generated error messages in the Apache error logs, has been added.
artf6001	Invalid TeamForge credentials entered during conversion to standalone Subversion Edge mode caused an exception. This is now fixed.

CollabNet Subversion Edge 1.3.3

Release Date: May 2011

Added features

Subversion Edge 1.3.3 adds these new features.

Artifact ID	Description
artf5449	To fix problems related to the 32 KB limit for an environment variable in Windows, TeamForge branding HTML is passed directly to ViewVC via Python.
artf5497	ViewVC in Subversion Edge has been upgraded to ViewVC 1.1.10.
artf5504	The copyright statements in Subversion Edge and ViewVC footers are updated for 2011.
artf5505	The Subversion Edge and ViewVC web interfaces include a proper attribution of the Apache Subversion trademarks.
artf5578	ViewVC in Subversion Edge supports the branding UI changes in TeamForge 6.1 -- primarily, the graphics for displaying broadcast messages.
artf5743	The post-install web page on Windows has been improved.
artf5745	The <code>svn_viewvc_httpd.conf</code> file adds support for a CollabNet TeamForge include file.
artf5746	In the <code>svn_viewvc_httpd.conf</code> file, the VirtualHost directives that were added for LDAP support are now conditional. If the Subversion Edge console is not using LDAP authentication mode (such as when it is managed by TeamForge), the VirtualHost directives are not specified.
artf5757	It is now easier to get to the Change Log view of binary (or a .sh) file. If the filename link does not go to the Change Log, then the link on the revision number does.
artf5907	Additional directives have been included when authenticating via LDAP.
artf5998	On Linux and Solaris, Subversion Edge 1.3.3 supports Subversion 1.6.17. Apache httpd has been updated to 2.2.19.

Resolved issues

Subversion Edge 1.3.3 resolves these issues.

Artifact ID	Description
artf5737	In a TeamForge server converted from Subversion Edge, clicking the Commit link for a file with a space in its name caused a 404 error. This is now fixed.

CollabNet Subversion Edge 1.3.2

Release Date: March 2011

Added features

Subversion Edge 1.3.2 adds these new features.

Artifact ID	Description
artf5359	The Apache configuration files include a comment on how users can edit the files if they need to.
artf5435	The Subversion Edge web console now creates secure session cookies. Jetty is configured to set the "secure" and "httpOnly" flags on cookies.
artf5509	Subversion binaries provided with Subversion Edge have been updated to version 1.6.16.

Resolved issues

Subversion Edge 1.3.2 resolves these issues.

Artifact ID	Description
artf5313	Editing user roles sometimes caused problems with the Subversion Edge toolbar. For example, the Administration tab intermittently disappeared for users with the ROLE_ADMIN role, after the ROLE_ADMIN role was removed for the admin user. This is now fixed.
artf5367	The Subversion Edge web console advertised the version for the Jetty application server. This has now been turned off.
artf5434	Subversion Edge allowed some of the webapp directories, including <code>csvn/</code> and <code>js</code> , to be browsed directly. Jetty is now configured so that these folders cannot be accessed.
artf5532	Subversion Edge did not encode error messages displayed for the Login Name and Email fields in the Create User page. This is now fixed.

CollabNet Subversion Edge 1.3.1

Release Date: November 2010

Added features

Subversion Edge 1.3.1 adds these new features.

Artifact ID	Description
artf5206	The Subversion binaries provided with Subversion Edge have been updated to version 1.6.15 .

Resolved issues

Subversion Edge 1.3.1 resolves these issues.

Artifact ID	Description
artf5114	When the SIGAR libraries didn't load, there was a faulty i18n message call which resulted in a 500 Internal Server Error. This is now fixed.
artf5173	The errors caused while editing a user account in a language other than English are now fixed.

CollabNet Subversion Edge 1.3

Release Date: October 2010

Added features

Subversion Edge 1.3 adds these new features.

Artifact ID	Description
artf4969	Subversion Edge 1.3 supports Solaris 10 SPARC/x86 systems.
artf4946	Internationalization is supported. Translations are available in Chinese, Japanese, Korean, and Portuguese. For more information, see this wiki .
artf4960	When Subversion Edge is configured to support LDAP, users can now log into the console using their LDAP credentials. They have ROLE_USER permissions, but can be assigned other roles to allow them to administer the server using their LDAP credentials.

Resolved issues

Subversion Edge 1.3 resolves these issues.

Artifact ID	Description
artf4906	The SSLSessionCache option for Apache is now included. You can see it in the SSL configuration section in [directory where you installed Subversion Edge]/data/conf/csvn_main_httpd.conf.
artf4911	The Grails code base has been upgraded from version 1.2.1 to version 1.3.4.
artf4924	Hyphens are allowed in repository names -- and the repositories can be used after conversion to TeamForge mode.
artf4937	Incorrect borders in the Repositories page have been fixed.
artf4944	Role and User lists no longer display the Id .
artf4945	In the Administration page, the Subversion Server Settings option has been renamed to Server Settings .
artf4959	ViewVC did not work when sudo option was used for starting Apache. This is now fixed.
artf4971	An incorrect message, "Your changes are not saved...", appeared when you switched between the General and Authentication tabs in Administration > Subversion Server Settings . This is fixed.

Artifact ID	Description
artf4979	The README is available in the root directory for Linux installs.
artf4980	If you enter an incorrect TeamForge server URL while trying to convert from standalone to TeamForge mode, you'll see get an error message to that effect. After this, if you try to click the "editIntegration" link, you'll see a message that you're in standalone mode but are trying to access a feature available only in TeamForge mode.
artf5003	Conversion to TeamForge failed when repository permissions were not validated. During the conversion process, you now get a warning message for repositories whose file permissions are not set correctly.
artf5004	During the conversion to TeamForge, usernames containing a dot (“.”) are not imported. A message appears when such usernames are identified, and these users will need to be created in TeamForge.
artf5013	Conversion to TeamForge mode now allows project names with dots.
artf5015	To support debugging, especially with problems during startup, the user-configured log level is set as early as possible.
artf5027	The server did not start when a firewall blocked packets on UDP port 5353 (the mDNS port used by Subversion Edge). This is now fixed.
artf5032	Performance related to gathering disk space statistics has been improved. Statistics gathering took a while to run on repositories with a lot of revisions. The job has been changed to run less frequently and completely asynchronous to the web UI.
artf5033	Subversion Edge 1.2.1 displayed incorrect values in graphs for available and used desk space. This is now fixed.
artf5037	When you change the Subversion Edge admin password, the Apache password is reset to the same. However, restarting the console reset the Apache password to admin again. This is now fixed.
artf5065	In the ViewVC version distributed with Subversion Edge 1.3, difflib is used if GNU diff is not available.
artf5067	The default port for the Apache server is set when you switch to or from SSL mode. For example, if you've specified port 80 currently and then select to use SSL with Apache, the port is changed to 443.
artf5068	For a repository listed in the Repositories page, click the corresponding Info link to see its statistics. In earlier versions, this link was called Edit .
artf5087	On some Windows systems, an incorrect warning about httpd.conf missing directives appeared. This is now fixed.
artf5095	<p>For some users, the Apache server did not start and no errors were logged. This was narrowed to these conditions:</p> <ul style="list-style-type: none"> • If LDAP was being used and the server was using a port < 1024 • If httpd was started by root or sudo <p>This is now fixed. You can start the server using <code>\$ sudo bin/csvn-httpd start</code></p>

Known Issues

The following issues are known to exist in Subversion Edge 1.3.

Artifact ID	Description
artf4961	This issue existed in previous releases as well, but the improved SSL support in this release makes it more likely you will run into this. When both the Subversion Edge console and the Apache server are configured for SSL, the first time you click on a link to the Apache server from the console, the browser does not connect to the server. This mainly seems to happen with Internet Explorer, although it might also happen with Chrome on Windows. Workaround: Clicking the link again works, as do direct bookmarks to the URL.

CollabNet Subversion Edge 1.2.2

Release Date: October 2010

Added features

Subversion Edge 1.2.2 adds these new features.

Artifact ID	Description
artf5055	Subversion binaries have been updated to version 1.6.13. For more information, see this post .

Resolved issues

Subversion Edge 1.2.2 resolves these issues.

Artifact ID	Description
artf5037	When you change the Subversion Edge admin user password, the Apache password is reset to the same. However, restarting the console reset the Apache password to "admin" again. This is now fixed.
artf5040	The logging of command line executions is now controlled so that the output is not logged when requested. Sensitive information, such as user passwords, doesn't appear in the logs.

CollabNet Subversion Edge 1.2.1

Release Date: September 2010

Resolved issues

Subversion Edge 1.2.1 resolves these issues.

Artifact ID	Description
artf4981	Service fails to start on some Windows systems due to failure to determine host address.

Artifact ID	Description
artf4990	Unexpected errors during startup are not allowed to prevent completion of bootstrap process.
artf5016	HTML characters in logs are escaped to prevent potential XSS or code injection.

CollabNet Subversion Edge 1.2

Release Date: September 2010

Added features

Subversion Edge 1.2 adds these new features.

Artifact ID	Description
artf63596	You can see the amount of disk space used by your Subversion repositories, including individual repositories, and how that usage has changed over time.
artf66797	Converting a Subversion Edge server to TeamForge mode is now supported on Windows systems.
artf71486	Customized <code>httpd.conf</code> files are backed up during the conversion to TeamForge mode. If the conversion fails, or if the TeamForge server is later reverted to standalone mode, the <code>httpd.conf</code> file is restored from the backup.
artf72110	Integration of the SIGAR library provides improved network and file system statistics.
artf4923	<ul style="list-style-type: none"> Subversion Edge is preconfigured for SSL on port 4434 using a self-signed certificate, in addition to the existing plain HTTP port 3343. Usage of SSL for the Subversion Edge console can be required by checking an option on the Administration tab. This causes requests on port 3343 to be redirected to SSL on port 4434.
	ViewVC fixes, including improvements to character encoding detection when viewing files.
	Pygments and Chardet Python libraries have been included to add color coding and character encoding support to ViewVC.

Resolved issues

Subversion Edge 1.2 resolves these issues.

Artifact ID	Description
artf71041	A repository is created in Subversion Edge only when the <code>svnadmin</code> repository creation operation succeeds.
artf71450	Conversion to TeamForge failed silently when a repository name in Subversion Edge was 128 characters long. The repository name can now contain a maximum of 32 characters, to accommodate TeamForge's requirement of a maximum of 128 characters for the repository path.
artf71565	A 404 error that occurred on clicking a link to an artifact, document, wiki or task in ViewVC has been removed.

Artifact ID	Description
artf71632	If the session with the TeamForge server times out during the conversion process, you see a more informative message. You would need to restart the server.
artf71843	A Null Pointer Exception during a failed conversion to TeamForge mode is fixed.
artf72901	A misleading message on creating or discovering repositories appeared after conversion to TeamForge mode. This has been removed.
artf73337	In the Administration > Subversion Server Settings page, the path to repositories entered by the user is normalized.
artf4904	a potential scalability issue in TeamForge's Subversion authorizer is fixed.

Known Issues

The following issues are known to exist in Subversion Edge 1.2.

Artifact ID	Description
artf4961	This issue existed in previous releases as well, but the improved SSL support in this release makes it more likely you will run into this. When both the Subversion Edge console and the Apache server are configured for SSL, the first time you click on a link to the Apache server from the console, the browser does not connect to the server. This mainly seems to happen with Internet Explorer, although it might also happen with Chrome on Windows. Workaround: Clicking the link again works, as do direct bookmarks to the URL.

CollabNet Subversion Edge 1.1.0

Release Date: July 2010

Added features

Subversion Edge 1.2 adds these new features.

Artifact ID	Description
artf51566	You can bring TeamForge features to Subversion Edge users on Windows by converting a standalone management console into an integration server in TeamForge.

Resolved issues

Subversion Edge 1.1.0 resolves these issues.

Artifact ID	Description
artf63552	ViewVC links in Status and Repositories pages open up in new windows or tabs.
artf65555	You get more informative error messages when Apache can't be started/restarted and during software updates.
artf65573	When you create a user, a value is required for email address.
artf69573	When installed as daemon, csvn did not start with the system (Linux). This is now fixed.

Artifact ID	Description
artf69961	On Linux systems where <code>sudo</code> was not installed, repeated warnings were logged in <code>console.log</code> . This is now fixed.
artf71166	<code>httpd.conf</code> and ViewVC/Python have been fixed to minimize noise in the error log.

CollabNet Subversion Edge 1.0.1

Release Date: July 2010

Resolved issues

CollabNet Subversion Edge 1.0.1 resolves these issues.

Artifact ID	Description
artf69136	CollabNet Subversion Edge 1.0.1 supports Subversion 1.6.12.
artf68877	An error that sometimes occurs when you select the Software Updates option, for example, when there is no Internet connection, has been fixed.
artf69277	Subversion Edge did not run on Linux if <code>sudo</code> was not installed. This is now fixed.
artf68780	A problem related to gathering network statistics that prevented some users from being able to log into Subversion Edge 1.0.0 on Windows XP, is fixed.
artf69286	" <code>csvn start</code> " reported a successful start when the <code>JAVA_HOME</code> environment variable was not set or set to an invalid path. This is now fixed.

CollabNet Subversion Edge 1.0.0

Release Date: June 2010

Highlights of Subversion Edge 1.0.0

- Subversion Edge runs on Microsoft Windows and Red Hat Enterprise Linux.
- Installing and updating are simple. For Windows, there is an installation wizard, and for Linux, it's a matter of unpacking a tar.gz file. Updates can be done from the web console.
- It is possible use a corporate LDAP repository to authenticate users.
- You can add new users via the web console. Users can also log in and edit their passwords that Apache uses for authentication.
- Role-based access control lets you assign roles to users to define their responsibilities.
- You can create new repositories as well as connect existing ones.
- You can maintain Subversion access control rules from the web console.
- Statistics such as network throughput and server disk space are available to monitor your server.
- Server logs are rotated daily and retained for the number of days you specify.
- It integrates the ViewVC repository browsing tool.
- The **openCollabNet** page provides a portal to a number of Subversion integrations, download clients and community forums supported by CollabNet.

Known Issues

The following issues are known to exist in CollabNet Subversion Edge 1.0.0.

Artifact ID	Description
artf66795	<p>When you install Subversion Edge on a Windows machine that has ActivePython 2.5 installed, you get a 500 Internal Server Error when you try to access ViewVC, and a corresponding "ImportError: DLL load failed" Python error in error.log.</p> <p>Workaround: To resolve the error, check for DLL's that start with "py" and move them from the system32 or SysWOW64 folder to a temp folder.</p>