

STEP System Solutions

# Patching STEP Trailblazer

**SiiboSystems**

Version 2.4

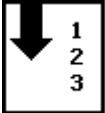
**Date:** February 28<sup>th</sup>, 2017

## Content

<b>Document Conventions .....</b>	<b>3</b>
<b>1 Introduction .....</b>	<b>4</b>
1.1 Selective Component Updates .....	4
1.2 Dependency Management .....	4
1.3 The STEP Ecosystem .....	4
<b>2 Patching Methods .....</b>	<b>5</b>
2.1 Direct Connection .....	5
2.2 Private Updates Mirror .....	6
2.2.1 Advantages .....	6
2.2.2 Requirements .....	6
<b>3 SPOT Program.....</b>	<b>8</b>
<b>4 Patching STEP Trailblazer.....</b>	<b>9</b>
4.1 Preface .....	9
4.2 Consider fallback procedure.....	9
4.3 Prepare.....	9
4.4 Install .....	10
4.5 Fallback.....	10
<b>5 Security .....</b>	<b>11</b>
<b>6 How to configure a private updates mirror .....</b>	<b>12</b>
6.1 Requirements .....	12
6.2 Ports used by the mirror server .....	12
6.3 iptables rules .....	13
6.4 Installation .....	15
6.5 Verifying the newly-configured mirror .....	16
6.6 Upgrading to new software .....	16
6.7 Starting without root and sysv init changes .....	16
6.8 Preemptive download .....	17
6.9 Limiting network bandwidth .....	17
6.10 Backing off in case of slow downloads .....	17

## Document Conventions

The following conventions are used throughout this document.

Convention	Description
Monospace text	Indicates commands and their options and also example output. Also used for configuration properties.
<i>Monospace italic text</i>	Indicates variables in commands.
<b>Monospace bold text</b>	Indicates text that must be entered from the keyboard.
<i>italics</i>	Indicates emphasis and documentation references.
	Steps to be executed by the person doing the installation. This person is referred to as the <i>STEP Installer</i> .

## 1 Introduction

---

In STEP Trailblazer, the STEP architecture has been split up into components and each component may access other components through a new set of component APIs. This componentized architecture satisfies otherwise contradictory requirements for longer time between releases and fast introduction of new improvements.

It will thus be possible to keep the core and other components unchanged for a longer time while choosing to upgrade selected components to take advantage of new features or important updates. The risk and workload involved in testing new updates is hence reduced.

### 1.1 Selective Component Updates

It is no longer necessary to upgrade to a completely new STEP release in order to take advantage of new components or features. If the feature is available in a new component, that component can simply be installed on its own. If the feature is available as an upgrade to an existing component, that component may simply be upgraded while keeping other components as they are.

Available component updates will be made visible on a STEP system similarly to the way updates are to mobile phone apps, i.e. with release notes detailing the new features and fixes available relative to the current installation and with instructions on how to perform the update.

### 1.2 Dependency Management

Components have separate release cycles limited only by the dependencies introduced when one component uses another component. Each component declares its dependency on other components through principles where a given component version may depend on a specified range of versions of another component.

### 1.3 The STEP Ecosystem

The Trailblazer release marks the inception of the STEP Ecosystem, where Stibo Systems and its partners will work together in developing extensions to the STEP platform that can be distributed to all STEP customers. Initially, the ecosystem will consist of the major STEP components, a long list of plugin/adaptor extensions to these components and various use case-specific components. The list of components and extensions will grow over time and businesses running STEP will be able to selectively install and use new components and extensions.

## 2 Patching Methods

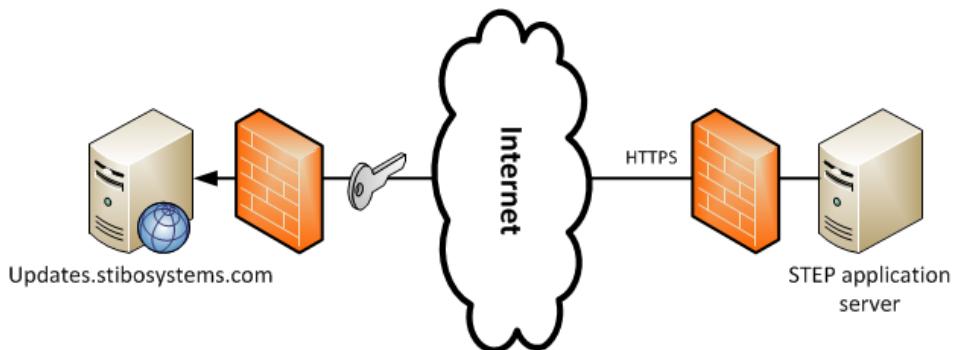
---

The installation and upgrade of new and existing components forms the patch operation of the STEP Trailblazer system; that operation will take place using one of two methods. Both methods will download the required software upgrade either from one of the Stibo Global Updates Mirrors (Release Server) or from an Private Updates Mirror at the customer. The connection to either of the two is using an encrypted network connection over HTTPS. Connection are always initiated from the customer side. The update mirror will at no time initiate a connection to the STEP Trailblazer environment.

The program responsible for downloading the required software upgrade and applying it to the STEP Trailblazer environment is named *STEP Patch Operation Tool* (SPOT). The SPOT program will always exist on the STEP Trailblazer application server.

### 2.1 Direct Connection

The default method for patching is the 'Direct Connection' method where the STEP Trailblazer environment is configured to directly allow an encrypted connection by HTTPS to the release server. This method offers the best security.

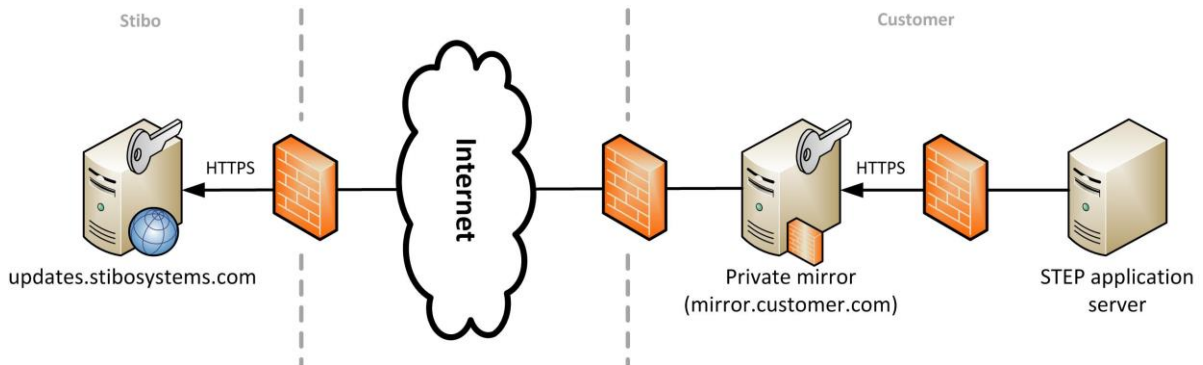


The advantages of using the 'Direct Connection' method are:

- ◆ Faster support from Stibo Systems by providing complete version information.
- ◆ Simpler infrastructure.

## 2.2 Private Updates Mirror

As an alternative to accessing the release server directly, it is possible to set up a Private Updates Mirror and configure SPOT on the internal STEP servers to use this mirror instead.



### 2.2.1 Advantages

The advantages of using the 'Private Updates Mirror' method are:

- ◆ If the Internet connection or the global updates server breaks down, then already downloaded files will still be available.
- ◆ The Internet connection bandwidth consumed is reduced by avoiding repeated downloads.
- ◆ The network configuration is simpler as only the mirror needs to access the updates server, while the individual SPOT instances can be configured to talk only to the private mirror on the internal network.

### 2.2.2 Requirements

To run a private mirror server, you need:

- ◆ A 64-bit Linux host, not shared with STEP.
- ◆ Java 8 64-bit (an updated version will be installed by SPOT, so the OS version is ok for bootstrapping.)
- ◆ Enough storage to hold the entire mirror, at the moment 400 GB will suffice.
- ◆ Outgoing Internet access to the Stibo updates servers on port 443.
- ◆ Incoming access from the private network on port 443 for the SPOT hosts .
- ◆ A DNS entry on the local network that can be expected to never change, so mirror.customer.com would be preferable to pc2016-02-13-room7-linux-test-dl120g9.dhcp.customer.com .

#### 2.2.2.1 Upstream root mirrors

The root mirrors that the private mirror connects to can be listed using `spot --mirrors`, but these are the current hosts:

- ◆ dk1.updates.stibosystems.com: Primary root mirror.
- ◆ dk2.updates.stibosystems.com: Secondary root mirror.
- ◆ updates.stibosystems.com: Fail-over mirror on a shared IP between the two root mirrors.

Outgoing TCP access on port 443 must be allowed to each of the root mirror IP addresses from the private mirror, this way the mirror has more upstream mirrors to pick from if one fails.

### 3 SPOT Program

---

The encrypted communication from the customer site to the update mirrors at Stibo Systems is always initiated by the SPOT program – be it on the STEP Trailblazer application server or on a dedicated SPOT support installation pc.

The communication sequence between the SPOT program and the update mirror is:

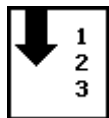
- ◆ SPOT stores the current thin snapshot of version information to updates.stibosystems.com .
- ◆ SPOT fetches the desired recipe of software bundles to download.
- ◆ SPOT downloads the actual bundles.
- ◆ SPOT stores the updated thin snapshot of version information to updates.stibosystems.com .

Storing the thin snapshots to updates.stibosystems.com serves two purposes:

1. Enable Stibo Systems to support the STEP Trailblazer environment by providing complete version information.
2. Enable easy creation of the exact software configuration for additional Test/QA environments and in the case of disaster recovery.

Both the metadata – including the thin snapshot – and the bundle recipe together with the actual bundles are cached by SPOT and only the files that are actually needed are ever downloaded, so the amount of data transferred is as low as possible.

The SPOT program is found in the home directory of the STEP Trailblazer installation on the application server. On a Linux server this will typically be in `/opt/stibo/step`. On a Windows server this will typically be in `E:\stibo\step`.



#### The SPOT program help menu

1. `cd /opt/stibo/step`
  2. `./spot --help`
-



## 4 Patching STEP Trailblazer

---

### 4.1 Preface

All commands listed are valid for any STEP environment, counting single application server setup and cluster.

### 4.2 Consider fallback procedure

#### Database:

Before patching STEP it should be considered how to do a fallback, if necessary. As a rule of thumb, the option to restore the database to a specific point in time should exist.

- ◆ Full backup of STEP database

Though, the requirement will depend on the actual patch. Please refer to the release note.

#### STEP application:

- ◆ Backup of and all files included in STEP\_HOME/config.properties
- ◆ Snapshot of STEP

How to do a snapshot of STEP

```
cd /opt/stibo/step
./spot --snapshot=/workarea/<snapshot-env-date>.spr
```

### 4.3 Prepare

The patch should be downloaded in advance to avoid unnecessary downtime for deployment.

```
./spot --prepare=to:step/trailblazer/step-trailblazer-<release>.spr
```

Above example if for a STEP core patch. Customers will potentially have their own components in addition and the command would look like:

```
./spot --prepare=to:step/trailblazer/step-trailblazer-<release>.spr,
to:customer/<customer>/<customer>-addon/7.0/<customer>-addon-7.0.x.spr
```

#### 4.4 Install

The patch should be installed by the following command:

```
./spot --apply=to:step/trailblazer/step-trailblazer-<release>.spr
```

With customer components:

```
./spot --apply=to:step/trailblazer/step-trailblazer-<release>.spr,  
to:customer/<customer>/<customer>-addon/7.0/<customer>-addon-7.0.x.spr
```

STEP will automatically stop and start during the patch session.

In case of any deprecated parameters in the configuration, please follow the instructions on the screen on how to correct and restart STEP

```
./spot --start
```

#### 4.5 Fallback

Depending on the contents of the patch the following steps should be completed for fallback:

1. Stop STEP

```
./spot --stop
```

2. Restore database (release note will tell if necessary)
3. Restore configuration files
4. Redeploy STEP using snapshot

```
./spot --apply=/workarea/<snapshot-env-date>.spr --sync --syncmode=delete
```

Using the snapshot and above '`--sync --syncmode=delete`' command will entirely recover STEP and delete any files related to a failed patch-session.

## 5 Security

---

Stibo Systems distributes software only via the `updates.stibosystems.com` server or one of the mirrors.

The update mirror web server is configured to communicate only HTTPS (never plain HTTP) on port 443, with only the high security cipher suites (using the Apache `SSLCipherSuite` "HIGH" option) and only communicate to clients having a proper client certificate issued by the build system certificate authority (CA) of Stibo Systems. This Stibo-specific CA was created solely for the purpose of certifying various STEP-related infrastructures.

Unlike a standard website where an external CA-signed certificate is used for ease of access by multiple clients (users), the updates server has only one client that is allowed to communicate with it and that is the SPOT client. For this reason, Stibo Systems believes this to be a safer and stronger security approach – over using an external CA certificate – as it is not possible for a cyberattacker to use a fake certificate from a compromised external CA to gain access.

By taking this approach, Stibo Systems realizes that it may cause some auditing tools to register a false positive and flag the server's certificate as self-signed. That being said, security teams or similar should configure these tools to trust Stibo Systems' CA to certify `stibosystems.com` domains.

Regarding the client certificate required for communicating with the update mirror, it is included in the STEP Trailblazer installation package and is used by the SPOT program to fetch both the software required for the initial installation and future application updates. Only the certificate used by the updates server will be trusted by SPOT for downloading these installation bits and updates.

All the certificates involved use 2048-bit RSA keys, so the system is considered secure against any man-in-the-middle attacker for the foreseeable future. Even with a valid client certificate, the operations allowed are severely limited to downloading only the licensed software produced by Stibo Systems and to saving customer-specific thin snapshots that do not contain software, so a compromised client would not be able to affect other customers or compromise other clients.

The SPOT program caches all files locally and validates contents using a SHA-1 hash before using the cached files, so the amount of traffic is kept as low as possible while ensuring the integrity of the cached files.

At no point will the STEP Trailblazer software communicate customer data back to the update mirrors at Stibo Systems. The thin snapshots uploaded to the release server contain only a list of versions of the installed STEP Trailblazer software components and they are only used by Stibo Systems to provide the best support to the STEP Trailblazer system.

## 6 How to configure a private updates mirror

---

### 6.1 Requirements

To run a private mirror server you need:

- ◆ A 64-bit Linux host, not shared with STEP.
- ◆ 64-bit Java 8 (an updated version will be installed by SPOT, so the OS version is okay for bootstrapping.)
- ◆ Enough storage to hold the entire mirror, at the moment 400 GB will suffice.
- ◆ Outgoing Internet access to the Stibo updates servers on port 443.
- ◆ Incoming access from the private network on port 443 for the SPOT hosts.
- ◆ A DNS entry on the local network that can be expected to never change, so `updates.example.com` would be preferable to `pc2016-02-13-room7-linux-test-dl120g9.dhcp.example.com`.

The root mirrors that the private mirror connects to can be listed using `spot --mirrors`, but these are the current hosts:

- ◆ `dk1.updates.stibosystems.com`: Primary root mirror.
- ◆ `dk2.updates.stibosystems.com`: Secondary root mirror.
- ◆ `updates.stibosystems.com`: Fail-over mirror on a shared IP between the two root mirrors.

Outgoing TCP access on port 443 must be allowed to each of the root mirror IP addresses from the private mirror, this way the mirror has more upstream mirrors to pick from if one fails.

### 6.2 Ports used by the mirror server

The mirror server listens on three ports:

- ◆ 10080: The Admin port of dropwizard, this is used to serve HTTP requests that allow monitoring the health of the server. The init script uses this port to check if the server is running.
- ◆ 10081: The stop port of Jetty, the init script uses this port to shut down the server in an orderly fashion. This port should not be accessed from outside the machine itself.
- ◆ 10082: The HTTPS service port that serves the actual mirror. This port should not be accessed from outside the machine itself.

These ports are all internal to the host that the server runs on and external systems should not connect directly to them (with the possible exception of having a monitoring system talking to port 10080.)



#### WARNING

Do not configure any STEP systems to talk to the mirror on port 10082. Port redirection (as described in the next section) must be set up.

### 6.3 iptables rules

The Java process is quite happy to run as an unprivileged user, but that makes listening to port 443 impossible, so to take care of that problem a set of iptables rules have to be used.

There are two ways to get the needed rules installed: Either run the mirror script as root when starting the server or set up iptables at the OS level.

If the init script is called by root, then it will take care of installing the needed port redirection, but if the administrator tasked with maintaining the mirror does not have sudo access to this script, then the rules can be inserted into the `/etc/sysconfig/iptables` config file, so the OS loads the rules at boot time.

These rules take care of redirecting all incoming requests to TCP port 443 over to port 10082 where the server listens.

To configure iptables on the server, switch to the **root** user and run the following command to view the current settings:

```
[root@mirror mirror]# /sbin/iptables -L -n --line-numbers
Chain INPUT (policy ACCEPT)
num target      prot opt source      destination
1  ACCEPT        all  --  0.0.0.0/0    0.0.0.0/0    state RELATED,ESTABLISHED
2  ACCEPT        icmp --  0.0.0.0/0    0.0.0.0/0
3  ACCEPT        all  --  0.0.0.0/0    0.0.0.0/0
4  ACCEPT        tcp  --  0.0.0.0/0    0.0.0.0/0    state NEW tcp dpt:22
5  REJECT        all  --  0.0.0.0/0    0.0.0.0/0    reject-with icmp-host-
prohibited

Chain FORWARD (policy ACCEPT)
num target      prot opt source      destination
1  REJECT        all  --  0.0.0.0/0    0.0.0.0/0    reject-with icmp-host-
prohibited

Chain OUTPUT (policy ACCEPT)
num target      prot opt source      destination
```

In the output, there will be a line that shows **REJECT** as the **INPUT** type, and in its first column (**num**), the line number is shown (**5** in the above example.) This line number will be the starting line for adding entries to the iptables configuration.

Once this information is known, run the following commands to add the needed port-opening entries:

```
[root@mirror mirror]# /sbin/iptables -I INPUT <line_number> -p tcp -m tcp --dport
443 -j ACCEPT
[root@mirror mirror]# /sbin/iptables -I INPUT <line_number> -p tcp -m tcp --dport
10082 -j ACCEPT
```

In the example above, the line number shown is 5, and therefore, the commands would look like this:

```
[root@mirror mirror]# /sbin/iptables -I INPUT 5 -p tcp -m tcp --dport 443 -j
ACCEPT
[root@mirror mirror]# /sbin/iptables -I INPUT 6 -p tcp -m tcp --dport 10082 -j
ACCEPT
```

Afterwards, add the entries for redirection by executing these commands:

```
[root@mirror mirror]# /sbin/iptables -t nat -A PREROUTING -p tcp -m tcp --dport
443 -j REDIRECT --to-ports 10082
[root@mirror mirror]# /sbin/iptables -t nat -A OUTPUT -o lo -p tcp -m tcp --dport
443 -j REDIRECT --to-ports 10082
```

Once that has been done, the added entries can be checked by running the commands that follow:

```
[root@mirror mirror]# /sbin/iptables -L -n --line-numbers
Chain INPUT (policy ACCEPT)
num target      prot opt source          destination
1  ACCEPT        all  --  0.0.0.0/0       0.0.0.0/0       state RELATED,ESTABLISHED
2  ACCEPT        icmp --  0.0.0.0/0       0.0.0.0/0
3  ACCEPT        all  --  0.0.0.0/0       0.0.0.0/0
4  ACCEPT        tcp  --  0.0.0.0/0       0.0.0.0/0       state NEW tcp dpt:22
5  ACCEPT        tcp  --  0.0.0.0/0       0.0.0.0/0       tcp dpt:443
6  ACCEPT        tcp  --  0.0.0.0/0       0.0.0.0/0       tcp dpt:10082
7  REJECT        all  --  0.0.0.0/0       0.0.0.0/0       reject-with icmp-host-prohibited

Chain FORWARD (policy ACCEPT)
num target      prot opt source          destination
1  REJECT        all  --  0.0.0.0/0       0.0.0.0/0       reject-with icmp-host-prohibited

Chain OUTPUT (policy ACCEPT)
num target      prot opt source          destination
```

```
[root@mirror mirror]# /sbin/iptables -L -n --line-numbers -t nat
Chain PREROUTING (policy ACCEPT)
num target      prot opt source          destination
1  REDIRECT      tcp  --  0.0.0.0/0       0.0.0.0/0       tcp dpt:443 redir ports 10082

Chain INPUT (policy ACCEPT)
num target      prot opt source          destination

Chain OUTPUT (policy ACCEPT)
num target      prot opt source          destination
1  REDIRECT      tcp  --  0.0.0.0/0       0.0.0.0/0       tcp dpt:443 redir ports 10082

Chain POSTROUTING (policy ACCEPT)
num target      prot opt source          destination
```

If everything looks to be correct, save the configuration so that it will be loaded each time the system reboots using these commands:

```
[root@mirror mirror]# /sbin/service iptables save
[root@mirror mirror]# /sbin/service iptables stop
[root@mirror mirror]# /sbin/service iptables start
```



#### IMPORTANT - For RHEL 7.x

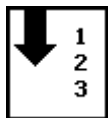
If the mirror server is a RHEL 7.x system, the above `/sbin/service iptables stop` and `/sbin/service iptables start` commands should be replaced with these instead:

```
/bin/systemctl stop iptables
/bin/systemctl start iptables
```

When things are all said and done, the `/etc/sysconfig/iptables` config file should look similar to this:

```
# Generated by iptables-save v1.4.7 on Tue Jun 21 14:16:10 2016
*nat
:PREROUTING ACCEPT [4595:497811]
:INPUT ACCEPT [1:28]
:OUTPUT ACCEPT [0:0]
:POSTROUTING ACCEPT [0:0]
-A PREROUTING -p tcp -m tcp --dport 443 -j REDIRECT --to-ports 10082
-A OUTPUT -o lo -p tcp -m tcp --dport 443 -j REDIRECT --to-ports 10082
COMMIT
# Completed on Tue Jun 21 14:16:10 2016
# Generated by iptables-save v1.4.7 on Tue Jun 21 14:16:10 2016
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [434:47393]
-A INPUT -m state --state RELATED,ESTABLISHED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 443 -j ACCEPT
-A INPUT -p tcp -m tcp --dport 10082 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited
COMMIT
# Completed on Tue Jun 21 14:16:10 2016
```

## 6.4 Installation



### Private mirror installation

1. Satisfy all requirements (see section 5.1 *Requirements* for details.)
2. Make a note of the DNS name which all the SPOT hosts will be using, in this example we will call it *mirror.customer.com*.
3. Contact Stibo Systems helpdesk and request that a system name is created for the mirror. This must be human readable and unique. System name in this example will be *your-mirror*.

Do not run any of these commands as `root`, make sure an unprivileged user exists for this mirror — e.g., `mirrorsw`.

4. Create a directory for the mirror.
5. Unzip the SPOT foothold (must be newer than the March 2016 release.)
6. Run: `./spot --enroll=mirror:your-mirror:mirror.customer.com`
7. Run: `./spot --apply=to:updates/mirror/latest.spr`
8. Edit the `mirror.yaml` file and review the options in the file, some of them, particularly the ones dealing with mailing of errors will need to be changed.
9. Run: `./mirror start`
10. Your mirror should now be running on `mirror.customer.com`.

- 
11. On a system with STEP installed, run this command:
 

```
./spot --updates=https://mirror.customer.com --ping
```
  12. As `root`, create a symlink to the mirror script into the appropriate `sysv` init directories using a command like this one:
 

```
ln -s <mirror_home>/mirror /etc/rc3.d/S90stibo-updates-mirror
```

E.g.,

```
ln -s /home/mirror/mirror /etc/rc3.d/S90stibo-updates-mirror
```
- 

## 6.5 Verifying the newly-configured mirror

Once the private mirror has been configured, it can be verified by running the following command on the STEP application server as the `stibosw` (or equivalent) user:

```
[stibosw@appl step]$ ./spot --mirrors
Stibo Patch Operations Tool
Priority Id      Name                               Url                               [X]
100 <customer> <customer> local mirror         https://mirror.customer.com      X
30  global     Auto failover mirror             https://updates.stibosystems.com
20  dk1        Primary mirror in Aarhus DK      https://dk1.updates.stibosystems.com
10  dk2        Secondary mirror in Aarhus DK    https://dk2.updates.stibosystems.com

Please use spot --mirrors --updates={Url} to set the upstream mirror
```

## 6.6 Upgrading to new software

The mirror can upgrade itself using the `init` script, simply run: `./mirror upgrade`  
 The upgrade command simply calls the `spot --apply=to:updates/mirror/latest.spr` and `./mirror restart` commands.

## 6.7 Starting without root and `sysv` init changes

If the `iptables`' rules have been added to the RHEL config file, the `init` script no longer needs `root` access and can be started by an unprivileged user by editing said user's `crontab` entries — i.e., `crontab -e` — and adding this line:

```
@reboot <mirror_home>/mirror start
```

E.g.,

```
@reboot /home/mirror/mirror start
```



## 6.8 Preemptive download

The mirror server is able to download files before the STEP systems ask for them. Doing this allows most files to be served from the local mirror without waiting for the upstream mirror, so better performance can be expected – at the cost of more disk space being utilized and the possibility of downloading files that end up never being needed.

The download option has three possible values:

- ◆ `download: HISTORIC`  
*Downloads all the files available from the upstream mirror, regardless of age, at the moment this requires about 1.5 TB of space.*
- ◆ `download: RELEASED`  
*Downloads newly released code as soon as it becomes available, this is the default and it will steadily consume space, at the moment about 2 GB is consumed per month.*
- ◆ `download: ON_DEMAND`  
*Nothing is downloaded until a client asks for it.*



### NOTE - Downloaded Content

When new content — e.g., monthly maintenance patches, add-on components, hotfixes, and similar — is downloaded to the mirror, it will be saved to the server's `<mirror_home>/content/takeout` directory (e.g., `/home/mirror/content/takeout .`)

## 6.9 Limiting network bandwidth

As no user is actively waiting for the preemptive downloads to complete and the downloads can be quite large, the bandwidth consumed by the background downloads can be limited via the `bulkDownloadSpeedInMbitPerSecond` configuration option.

The default limit is 10 Mb/s, so the expected lag after a release of STEP until the mirror is in sync should be less than an hour.

The bulk download speed limit is applied to the download of newly released files and historically released files separately, so if a historic download is running, then the two bulk processes can consume twice the speed limit in total.

## 6.10 Backing off in case of slow downloads

If downloads take a long time to complete, then it could be because the network or the upstream mirror is overloaded, so to avoid contributing to the problem the bulk download threads will back off (sleep) for a while after completing a download.

The amount of time to sleep after a download can be specified using the `bulkBackoffFactor` option, which defaults to 1.5 .

For example, if a download takes 2 seconds then a `bulkBackoffFactor` of 1.5 means that the process will sleep 3 seconds before downloading the next file.

## About Stibo Systems

Stibo Systems provides global organisations with a leading multi-domain Master Data Management (MDM) solution. Stibo Systems enables its customers to better manage enterprise intelligence on a global scale, improve sales, and quickly adjust to changes in business requirements. Stibo Systems' STEP technology is a flexible MDM solution that provides a single trusted source of operational information for the entire enterprise. Stibo Systems offers industry-specific solutions, engineered and supported to meet the strategic information needs of global customers including: GE, Sears, Siemens, Target and Thule. Stibo Systems is a subsidiary of the privately held Stibo A/S group, originally founded in 1794 with corporate headquarters in Aarhus, Denmark.

For more information, please visit [www.stibosystems.com](http://www.stibosystems.com) .