

# Software Update API. v1.0

Copyright © Riverbed Technology Inc. 2024

Created Jan 16, 2024 at 02:01 PM

# Overview

## Overview

The documentation pages in this section describe the RESTful APIs included with Riverbed SteelCentral Products. It is assumed that the reader has practical knowledge of RESTful APIs, so the documentation does not go into detail about what REST is and how to use it. Instead the documentation focuses on how iso image can be uploaded, how to track progress for iso upload, removing iso and all the software update related operations.

The following operations can be performed via the API:

- Upload or Fetch .iso image to Riverbed SteelCentral.
- Track progress of .iso image upload to Riverbed SteelCentral.
- Delete .iso images using image id from Riverbed SteelCentral.
- Track progress of .iso image upload.
- Track progress of Software Update on Riverbed SteelCentral.
- Initialize Software Update on Riverbed SteelCentral.
- Uninitialize Software Update on Riverbed SteelCentral.
- Start Software Update on Riverbed SteelCentral.

Details about REST resources can be found in the **Resources** section. This overview continues with how to run these API's.

## Authentication

All REST requests must be authenticated. The **Authentication** section of the Common API describes which authentication methods are presently supported.

## Resources

### Images: Upload ISO

The .iso image needs to be passed as a binary object using form-data where key = 'file' and value = 'file object'

```
PUT https://{device}/api/mgmt.system_update/1.0/images/upload
```

#### Authorization

This request requires authorization.

#### Request Body

Do not provide a request body.

#### Response Body

On success, the server does not provide any body in the responses.

### Images: Fetch ISO

Fetches ISO from a specified url.

```
PUT https://{device}/api/mgmt.system_update/1.0/images/fetch
```

#### Authorization

This request requires authorization.

#### Request Body

Provide a request body with the following structure:

```
JSON
```

```
{
  "url": string
}
```

Example:

```
{
  "url": "http://xxxx:xxxx/install-10.21-xxx-192.iso"
}
```

Property Name	Type	Description	Notes
<i>FetchImage</i>	<object>	Fetches .iso file from specified url to the Riverbed SteelCentral device.	
<i>FetchImage.url</i>	<string>	URL from where ISO needs to be fetched to Riverbed SteelCentral device.	

## Response Body

On success, the server does not provide any body in the responses.

## Images: Get ISO Image Details

Provides Information about ISO Images Uploading / Uploaded.

```
GET https://{device}/api/mgmt.system_update/1.0/images
```

## Authorization

This request requires authorization.

## Response Body

On success, the server returns a response body with the following structure:

JSON

```
[
  {
    "id": string,
    "filename": string,
    "progress": string,
    "state": string,
    "checksum": string,
    "version": string
  }
]
```

Example:

```
[
  {
    "id": 1,
    "filename": "install-10.22-xxxx-192.iso",
    "progress": "100.0%",
    "state": "Uploaded",
    "checksum": "94b58c6a41f18605705a43c23d4859991f91303337bdc412f9e53d2ca9758484",
    "version": "10.22"
  }
]
```

Property Name	Type	Description	Notes
<i>Images</i>	<array of <object>>	Information about ISO Images Uploading / Uploaded.	
<i>Images[Image]</i>	<object>	ISO image details .	Optional
<i>Images[Image].id</i>	<string>	A unique identifier for the ISO image.	
<i>Images[Image].filename</i>	<string>	Name of the ISO image.	
<i>Images[Image].progress</i>	<string>	The progress of the current upload/fetch operation.	
<i>Images[Image].state</i>	<string>	The state of the ISO image getting uploaded.	Values: Uploaded, Uploading, Upload Error
<i>Images[Image].checksum</i>	<string>	The SHA-256 checksum of the uploaded ISO image.	
<i>Images[Image].version</i>	<string>	The version of the uploading/uploaded ISO image.	

## Images: Get ISO Image Details using id

Provides Information about a particular ISO Image uploading / uploaded.

```
GET https://{device}/api/mgmt.system_update/1.0/images/items/{id}
```

## Authorization

This request requires authorization.

## Response Body

On success, the server returns a response body with the following structure:

### JSON

```
{  
  "id": string,  
  "filename": string,  
  "progress": string,  
  "state": string,  
  "checksum": string,  
  "version": string  
}
```

Example:

```
{  
  "id": 1,  
  "filename": "install-10.22-xxxx-192.iso",  
  "progress": "51%",  
  "state": "Uploading",  
  "checksum": "94b58c6a41f18605705a43c23d4859991f91303337bdc412f9e53d2ca9758484",  
  "version": "10.22"  
}
```

Property Name	Type	Description	Notes
<i>Image</i>	<object>	Information about a particular ISO Image Uploading / Uploaded.	
<i>Image.id</i>	<string>	A unique identifier for the ISO image.	
<i>Image.filename</i>	<string>	Name of the ISO image.	
<i>Image.progress</i>	<string>	The progress of the current upload/fetch operation.	
<i>Image.state</i>	<string>	The state of the ISO image getting uploaded.	Values: Uploaded, Uploading, Upload Error
<i>Image.checksum</i>	<string>	The SHA-256 checksum of the uploaded ISO image.	
<i>Image.version</i>	<string>	The version of the uploading/uploaded ISO image.	

## Images: Delete ISO Image

Removes the ISO and clears Information about the same.

```
DELETE https://{device}/api/mgmt.system_update/1.0/images/items/{id}
```

## Authorization

This request requires authorization.

## Response Body

On success, the server does not provide any body in the responses.

## Update: Update Status

Get the status of Software Update.

```
GET https://{device}/api/mgmt.system_update/1.0/update
```

## Authorization

This request requires authorization.

## Response Body

On success, the server returns a response body with the following structure:

## JSON

```
{
  "State": string,
  "Internal State description": string,
  "State description": string,
  "Current Version": string,
  "Update To Version": string
}
```

Example:

```
{
  "State": "neutral",
  "Internal State Description": "Visible only during State = failed_graceful or failed_critical",
  "State Description": "Visible only during State = failed_graceful or failed_critical",
  "Current Version": "10.22",
  "Update To Version": "10.23"
}
```

Property Name	Type	Description	Notes
<i>UpdateStatus</i>	<object>	Software Update Current Status.	
<i>UpdateStatus.State</i>	<string>	The state of the update.	Values: neutral, initializing, initialized, running, failed_graceful, failed_critical
<i>UpdateStatus.Internal State description</i>	<string>	Displays Internal State Description only at the time of failure during software update.	Optional
<i>UpdateStatus.State description</i>	<string>	Displays State Description only at the time of failure during software update.	Optional
<i>UpdateStatus.Current Version</i>	<string>	Displays Current Riverbed SteelCentral version.	
<i>UpdateStatus.Update To Version</i>	<string>	Displays Target version to which Riverbed SteelCentral needs to be updated to.	

## Update: Initialize

Initialize the Riverbed SteelCentral. The operation is synchronous.

```
POST https://{device}/api/mgmt.system_update/1.0/update/init
```

### Authorization

This request requires authorization.

### Request Body

Do not provide a request body.

### Response Body

On success, the server does not provide any body in the responses.

## Update: Uninitialize

Uninitialize the Riverbed SteelCentral. The operation is synchronous.

```
POST https://{device}/api/mgmt.system_update/1.0/update/reset
```

### Authorization

This request requires authorization.

### Request Body

Do not provide a request body.

### Response Body

On success, the server does not provide any body in the responses.

## Update: Start Update

Start the software update on the Riverbed SteelCentral. The operation is asynchronous

```
POST https://{device}/api/mgmt.system_update/1.0/update/start
```

## Authorization

This request requires authorization.

## Request Body

Do not provide a request body.

## Response Body

On success, the server does not provide any body in the responses.

## Error Codes

In the event that an error occurs while processing a request, the server will respond with appropriate HTTP status code and additional information in the response body:

```
{
  "error_id": "{error identifier}",
  "error_text": "{error description}",
  "error_info": {error specific data structure, optional}
}
```

The table below lists the possible errors and the associated HTTP status codes that may returned.

Error ID	HTTP Status	Comments
PARTIAL_CONTENT	206	Operation was partially successfull.
BAD_REQUEST	400	Invalid argument.
AUTH_REQUIRED	401	Authentication required.
AUTH_INVALID_CREDENTIALS	401	Make sure the user name is correct and type your password again. Be sure to use the correct case for the user name and password.
INTERNAL_ERROR	500	Internal error.