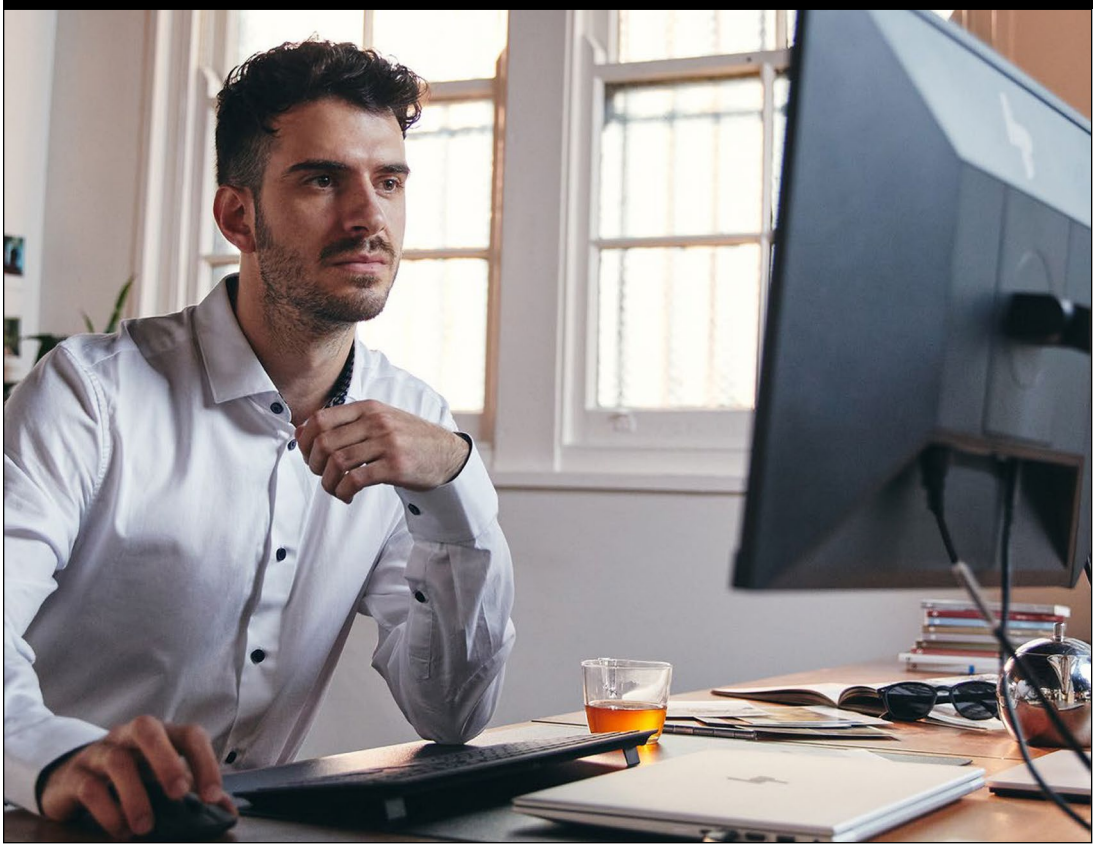
Technical Whitepaper

# HP Remote System Controller's Redfish API

Contents & Navigation


| | |
|---|---|
| Redfish | 3 |
| Why Redfish? | 3 |
| Authentication | 4 |
| Updating the RSC | 5 |
| Handling Tasks | 6 |
| Configuring Virtual Media | 7-8 |
| Changing Host BIOS Settings | 8 |

## Purpose of this Document

HP Remote System Controller's (RSC) API implements the DMTF's Redfish® standard for manageability. While DMTF's Redfish is documented very well, with white papers (https://redfish.dmtf.org/education/whitepapers), specification files (https://www.dmtf.org/standards/redfish), and well-documented schemas (https://redfish.dmtf.org/schemas), there are nuances to the RSC API implementation. This document will help describe the different areas that need more information than our Swagger Docs and the Redfish documentation can provide. For general documentation of the API endpoints and resources the Swagger Docs (https://developers.hp.com/hp-remote-system-controller/api/hp-remote-system-controller-api-2308) can provide the best information.

Technical Whitepaper

Contents & Navigation


Redfish 3

Why Redfish? 3

Authentication 4

Updating the RSC 5

Handling Tasks 6

Configuring Virtual Media 7-8

Changing Host BIOS Settings 8


# Table of Contents


**Redfish** ..... 3

**Why Redfish?** ..... 3

**Authentication** ..... 4

- Default Password ..... 4
- Changing the Password ..... 4
- Session Restrictions ..... 4

**Updating the RSC** ..... 5

- SimpleUpdate ..... 5
- MultipartPush ..... 5
- Automatic Updates ..... 5
- Downgrade Protection ..... 5

**Handling Tasks** ..... 6

- Task Monitoring ..... 6
- Cancelling Tasks ..... 6
- Update Tasks ..... 6

**Configuring Virtual Media** ..... 7

- Downloading Image File To RSC ..... 7
- Uploading Image File to RSC ..... 7
- Clearing Image Data ..... 7
- Inserting and Ejecting ..... 7
  - •Insert ..... 7
  - •Eject ..... 8

**Changing Host BIOS Settings** ..... 8

Technical Whitepaper

Contents & Navigation


| | |
|---|---|
| **Redfish** | 3 |
| **Why Redfish?** | 3 |
| **Authentication** | 4 |
| **Updating the RSC** | 5 |
| **Handling Tasks** | 6 |
| **Configuring Virtual Media** | 7-8 |
| **Changing Host BIOS Settings** | 8 |


# Redfish

Redfish is the modern RESTful protocol for remote management of workstations, servers, storage, networking, and converged infrastructure. HP Inc. is a Promoter member (https://www.dmtf.org/standards/spmf) of the Redfish Forum, helping steer the standard in ways that benefit our customers and products.

Redfish is the replacement for IPMI as the industry standard for server manageability. Server fleet management is now mostly done through Redfish, with IPMI supported for legacy solutions.

By implementing the same Redfish standard protocol, the HP Remote System Controller can be used to manage workstations via the same fleet management and client solutions already used for servers. This provides enhanced mixed fleet support for customers.

The Redfish standard is designed to be expandable by OEMs, the implementors of the standard. Each resource can add `OEM` tags and actions to extend the protocol's base functionality.

# Why Redfish?

The HP Remote System Controller solution deliberately chose Redfish as the standard on which to base the API due to the following reasons:

- Redfish is a modern standard showing great promise in becoming a future foundational management solution for all manner of devices, not just servers and traditional datacenter solutions.
- Redfish implements some of the latest security standards and best practices, and as a Promoter Member, HP intends to extend those security features to cover all manner of deployments and potentially hostile environments.
- HP computers are being deployed alongside servers in datacenters, data closets, and in non-traditional environments, and for our customers, having a common manageability strategy is critical to achieving operational efficiency.
- Entire workflows can be scripted out. This will become even more powerful as we make more and more out-of-band operations automated through the API. This allows for scenarios where hardware is installed, and then the rest of the deployment workflow is handled remotely, where at the end of the workflow your entire fleet is ready for end users to be productive.

HP will continue to invest in building out our Redfish-based API and evolve the RSC solution based on customer needs. If you have specific needs that can be met by Redfish schemas that we do not currently support, we want to know about it. Please contact your HP Sales Representative to make requests for Redfish functionality that you need.

Technical Whitepaper

Contents & Navigation


| | |
|---|---|
| Redfish | 3 |
| Why Redfish? | 3 |
| **Authentication** | 4 |
| Updating the RSC | 5 |
| Handling Tasks | 6 |
| Configuring Virtual Media | 7-8 |
| Changing Host BIOS Settings | 8 |


# Authentication

The RSC API only supports Session based authentication. Creating a session is done by passing the username and password with a POST to the `/redfish/v1/Sessions` endpoint. The POST will return an `x-auth-token` header and a location header containing the URI of the session (in the form of `/redfish/v1/Sessions/{id}`). The `x-auth-token` header must be sent with every subsequent request to the RSC API.

For security, the RSC API does not support Basic Authentication, and HP will continue to adopt the best security practices and offerings that the Redfish specification has to offer, in addition to promoting new security protocols as available.

# Default Password

Each RSC device has a default password, which is stored in the onboard TPM. This default password is printed on the RSC itself. The default password must be changed before any additional calls to the API can be performed. The default password can be used to create a Session, but the session is only valid for a PATCH to the `/redfish/v1/AccountService/Accounts/{id}` to change the password. On a factory reset of the RSC, the administrator password is reset to the default password for the device.

It is recommended that the default password be stored securely upon initial installation of the hardware, such that even if the labeling is damaged, you have a record of the default password to remotely restore access to the RSC. A QR code is included on each label to make capturing the serial number, default password, and MAC address easier, while reducing the chance for human error in recording the information.

# Changing the Password

To change an account password, perform a PATCH on the `/redfish/v1/AccountService/Accounts/{id}` endpoint, passing in the `Password` parameter with the new password. To protect against session hijacking attacks there is an additional restriction in place for the RSC API. The PATCH to the account resource must be the only request made during a Session. If any request is made before the PATCH, a new Session is required to be created, validating the credentials again.

# Session Restrictions

The RSC sessions will expire after 1 hour of inactivity. Each time the session is used to perform a request to the API the session is extended another hour. Once the session expires the user will be required to log in again by creating a new session.

All RSC sessions will expire after 8 hours, regardless of activity.

Technical Whitepaper

Contents & Navigation


| | |
|---|---|
| Redfish | 3 |
| Why Redfish? | 3 |
| Authentication | 4 |
| **Updating the RSC** | 5 |
| Handling Tasks | 6 |
| Configuring Virtual Media | 7-8 |
| Changing Host BIOS Settings | 8 |


# Updating the RSC

The RSC firmware can be updated remotely in a variety of ways. The update process utilizes an Update Package containing the new firmware which is either downloaded from or uploaded to an RSC device. Once the Update Package is located on the RSC device the update process begins. The update package is extracted, and all hashes and signatures of the contents are verified. Only once all components are verified will the update begin.

The RSC implements the Redfish standard SimpleUpdate and MultipartPush update operations, in addition to automatic updates. RSC firmware update packages can be downloaded from the following website, where new update packages are expected to be released on roughly a quarterly basis: https://rsm.hp.com/console/download/firmware

# SimpleUpdate

SimpleUpdate is an update operation which downloads an Update Package from a URI and updates the RSC firmware. The simple update is initiated with a POST to the `/redfish/v1/UpdateService/Actions/SimpleUpdate` endpoint. The body of the POST must contain the `ImageURI` parameter which points to the update package. The RSC will begin by downloading the update package from the ImageURI and then the update process continues.

# MultipartPush

MultipartPush is the update operation that uploads an Update Package from the client device to the RSC device. This requires the update package be on the file system of the client device making the call. MultipartPush follows the multipart/ form-data (rfc7578)[https://www.ietf.org/rfc/rfc7578.txt] spec. The POST requires the `UpdateFile` form-data which is of type `application/ octet-stream` and contains the Update Package. When the operation begins the RSC will stream the data to its file system and then the update process continues once all data is received.

# Automatic Updates

Automatic updates are available for RSC devices with internet access. Automatic updates are disabled by default. To enable perform a PATCH on `/redfish/v1/ UpdateService` with the body `{Oem:{HP:{“AutoUpdateEnabled”: true}}}`. When enabled the RSC will check for updates once a week, based on when the Automatic Update setting was enabled, or when the RSC reboots. The RSC will reach out to the HP Remote System Management (RSM) cloud servers (rsm.hp.com) and request information about any possible updates, if an update exists for the device it will perform a download and update similar to the SimpleUpdate process. The update package is downloaded from the HP RSM cloud directly, without needing to pass this information to the device.

# Downgrade Protection

The RSC device prevents any downgrade of FW to ensure security. A factory reset of the device will not revert any installed FW or SW on the device.

Technical Whitepaper

Contents & Navigation


| | |
|---|---|
| Redfish | 3 |
| Why Redfish? | 3 |
| Authentication | 4 |
| Updating the RSC | 5 |
| **Handling Tasks** | 6 |
| Configuring Virtual Media | 7-8 |
| Changing Host BIOS Settings | 8 |


# Handling Tasks

Redfish has defined how tasks shall be used and handled in a service that implements the protocol. When any request is not able to complete its operation and return right away a Task resource and, a 202 status code shall be returned. The Task resource holds the state of the task.

## Task Monitoring

The Task's `TaskMonitor` parameter contains a URI that can be used to monitor the Task. This is a unique endpoint that behaves differently than any other endpoints in Redfish. When a GET is performed on the TaskMonitor URI the response can be either:

- A 202 status code with the same Task resource. This indicates that the task is still in progress.
- The response that would have been returned by the operation if it was instantly completed. As examples:
    - If a Task to power on the Host was completed successfully; the TaskMonitor would return a 200 code.
    - If a task to perform an update fails to verify the required signatures; the TaskMonitor would return a 400 code with a RedfishError, explaining the failure reasons in the body.

## Cancelling Tasks

Some tasks can be canceled. To cancel a task a DELETE operation can be called on either the Task resource URI itself or the `TaskMonitor` URI. On a successful cancellation the request will return a 204 code, indicating the deletion. If the task does not support cancelation the request will return a 405 code.

Only a select few tasks can be interrupted and canceled, due to the operations being performed. In the 23.08.0 version of the Remote System Controller only the enrollment to the Remote System Management server can be cancelled.

## Update Tasks

Tasks that are returned while performing a manual update have additional tracking information to monitor the update process. The Task resource contains a `PercentComplete` parameter which is the integer representation of the update's completion percentage. The Task resource also contains a `Messages` parameter which is an array of RedfishMessages. If the update is still in progress the current stage of the update process will be indicated via the first message in the array.

Technical Whitepaper

Contents & Navigation


| | |
|---|---|
| Redfish | 3 |
| Why Redfish? | 3 |
| Authentication | 4 |
| Updating the RSC | 5 |
| Handling Tasks | 6 |
| **Configuring Virtual Media** | 7-8 |
| Changing Host BIOS Settings | 8 |


# Configuring Virtual Media

The RSC device supports a single virtual media image that configured via the `/redfish/v1/Systems/1/VirtualMedia/1` endpoint. The image data is required to be stored on the RSC device locally to be exposed to the Host as a CDROM, making the Virtual Media read only, to ensure that user data (only pixels via KVM) cannot leave the host via the RSC. The image can be inserted and ejected from the Host. The image has a max size of 4.7 GBs (the capacity of a standard DVD) and is non-volitile, such that you can store imageswith tools, installers, or in other useful files for immediate use by the host platform.

## Downloading Image File To RSC

The image data can be downloaded to the RSC by performing a PATCH on `/redfish/v1/Systems/1/VirtualMedia/1` and setting the `Image` property to the URL that holds the data. The RSC will then download the data and return a task to track the download. The virtual media resource must not be inserted and, while the download is in progress, the image will be locked during the process.

## Uploading Image File to RSC

Much like the update package, the Virtual Media Image can be pushed to the RSC via a multipart/form-data (rfc7578)[https://www.ietf.org/rfc/rfc7578.txt] spec call. This operation is not currently part of the Redfish standard, but was implemented for ease of use.

To start the upload perform a multipart POST to `/redfish/v1/Systems/1/VirtualMedia/1/MultipartImage`. The POST requires the ` ImageFile` formdata which is of type `application/octet-stream` and contains the image file. The virtual media resource must not be inserted and while the upload is in progress the image will be locked during the process.

The filename of the uploaded image will be stored in the `Oem->HP->CustomTag` parameter.

## Clearing Image Data

To clear the stored image perform a PATCH to `/redfish/v1/Systems/1/VirtualMedia/1` setting the `Image` properly to either `null` or `””`.

## Inserting and Ejecting

Inserting and Ejecting the image as a CDROM to the Host can be done by either PATCH operations or by Redfish Actions with POST operations.

### Insert

Inserting the image can be done by a PATCH changing the `Inserted` property to `true`. It can also be done by performing a POST to `/redfish/v1/Systems/1/VirtualMedia/1/Actions/VirtualMedia.InsertMedia`. If using the InsertMedia action the body can either be empty or contain an `Image` property. If the `Image` is passed it will update the image data before inserting the virtual media, returning a task to the POST.

Contents & Navigation


| | |
|---|---|
| Redfish | 3 |
| Why Redfish? | 3 |
| Authentication | 4 |
| Updating the RSC | 5 |
| Handling Tasks | 6 |
| Configuring Virtual Media | 7-8 |
| **Changing Host BIOS Settings** | 8 |


## Eject

Ejecting the image can be done by a PATCH changing the `Inserted` property to `false`. It can also be done by performing a POST to `/redfish/v1/Systems/1/VirtualMedia/1/Actions/VirtualMedia.EjectMedia`.

# Changing Host BIOS Settings

Following the Redfish standard, the RSC device supports changing Host BIOS settings. The RSC device only supports applying the BIOS settings on the next reboot of the Host. To view the current Host BIOS settings perform a GET on `/redfish/v1/Systems/1/Bios`. The BIOS settings will be listed in `Attributes` parameter. To change the Host BIOS settings the `/redfish/v1/Systems/1/Bios/Pending` can be used. The pending BIOS settings endpoint can also be found in `@Redfish.Settings->SettingsObject->odata.id`, the same as other Redfish implementations. To get the most out of this feature, it is recommended to update your hosts to the latest BIOS version, as available settings will be phased in across future BIOS updates.

To change the pending settings perform a PATCH on `Attributes` property on the `/redfish/v1/Systems/1/Bios/Pending` endpoint. The pending BIOS changes can be viewed at any time by performing a GET on `/ redfish/v1/Systems/1/Bios/Pending `.

To apply the changes, reboot the Host.

## Let us help you create amazing business solutions today

LEARN MORE



4AA8-3272ENW, April 2024